Learning-based Segmentation Framework for Tissue Images Containing Gene Expression Data

Musodiq Bello *Student Member, IEEE*, Tao Ju, James Carson, Joe Warren, Wah Chiu,
Ioannis A. Kakadiaris *Member, IEEE*

***Abstract*— Associating specific gene activity with functional locations in the brain results in a greater understanding of the role of the gene. To perform such an association for the over 20,000 genes in the mammalian genome, reliable automated methods that characterize the distribution of gene expression in relation to a standard anatomical model are required. In this paper, we propose a new automatic method that results in the segmentation of gene expression images into distinct anatomical regions in which the expression can be quantified and compared with other images. Our contribution is a novel hybrid atlas that utilizes a statistical shape model based on a subdivision mesh, texture differentiation at region boundaries, and features of anatomical landmarks to delineate boundaries of anatomical regions in gene expression images. This atlas, which provides a common coordinate system for internal brain data, is being used to create a searchable database of gene expression patterns in the adult mouse brain. Our framework annotates the images about four times faster and has achieved a median spatial overlap of up to 0.92 compared with expert segmentation in 64 images tested. This tool is intended to help scientists interpret large-scale gene expression patterns more efficiently.**

***Index Terms*— Segmentation, texture classification, shape modeling, feature selection, gene expression.**

I. Introduction

AFTER sequencing several mammalian genomes typically consisting of over 20,000 genes [1], the next challenge facing the biomedical community is to determine the function of these genes. Knowledge of gene function is important for a better understanding of diseases and the development of new therapies. By relating a gene's activity to specific functional populations of cells, much insight is gained into the function of the gene's products. In addition, genes that express in the same set of cell populations are likely to be related in function. The mouse is a well-established model system for exploring gene function and disease mechanisms. Consequently, determining where genes are active in different mouse tissues can lead to a greater understanding of how gene products affect human diseases.

One strategy towards this end is high-throughput *in situ* hybridization (HITISH) [2], which is a histological method that can be applied to reveal cellular-resolution gene expression in tissue sections rapidly [3]. Using HITISH, the spatial distribution of gene expression patterns (where genes are actively transcribed) can be generated throughout an entire organ or organism for thousands of genes. *In situ* hybridization (ISH) is currently being applied to both the mouse brain and the mouse embryo [4], [5], [6]. In this work, we focus on the postnatal mouse brain which is small, accessible, and contains hundreds of functional subregions. Such characteristics make it a particularly good specimen choice for HITISH. The study of gene expression patterns at cellular resolution is appropriate for addressing questions about the role of genes in cell identity, differentiation, and signaling. By organizing a large collection of gene expression patterns into a digital atlas, HITISH data can be used to make significant advances in functional genomics, as DNA sequence databases have already done.

Accurately annotating the location of gene expression in relation to the underlying anatomical morphology is a prerequisite to the interpretation of HITISH results. Gene expression images must be segmented into regions and subregions in which expression patterns can be compared across images. Such segmentation/annotation must become both rapid and accurate in order to handle the vast amount of data generated by HITISH. A major computational challenge is to develop efficient algorithms that can accomplish this task with little or no human intervention.

In this paper, we propose a new automatic method that results in the segmentation of gene expression images into distinct anatomical regions and subregions using a statistical hybrid atlas based on subdivision surfaces. The atlas utilizes a statistical shape model, texture differentiation at region boundaries, and features of selected anatomical landmarks to delineate anatomical region boundaries. This atlas provides a common coordinate system for internal brain data through which gene expression patterns can be compared across images.

The rest of the paper is organized as follows: Section II presents related work and background information about the application domain. The details of our proposed learning-based segmentation framework are described in Section III. The training of the model is described in Section IV while the deployment is presented in Section V. The results are discussed in Section VI.

Manuscript received July 20, 2006; revised December 1, 2006. The authors thank Christina Thaller and Gregor Eichele their help with data acquisition and annotation. This work was supported in part by a training fellowship from the W.M. Keck Foundation to the Gulf Coast Consortia through the Keck Center for Computational and Structural Biology, a NuView Foundation scholarship, NLM Grant 5T15LM07093 from the Keck Center for Computational and Structural Biology, P41 RR02250 grant from the National Center for Research Resources, and DOE LDRD DE-AC05-76RL01830.

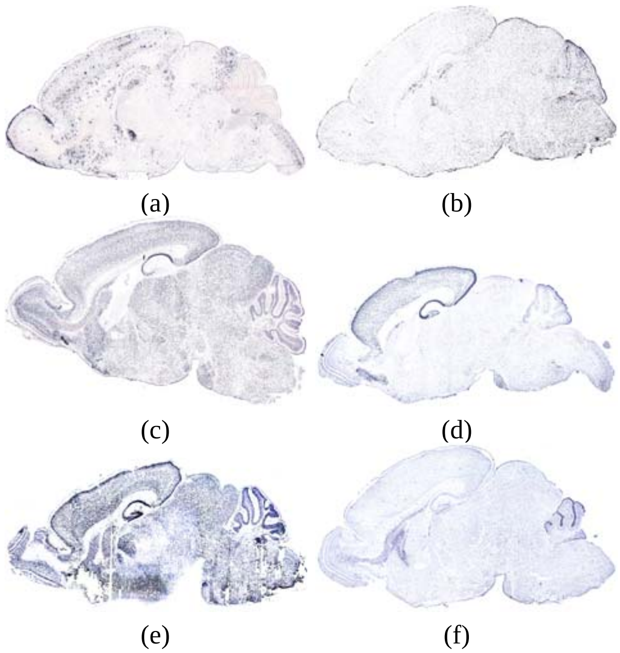


Fig. 1. Sample images in the same cross-section level showing (a,b) variation in expression patterns; (c, d) variation in shape; (e,f) presence of artifacts and missing parts.

II. Gene Expression Data

As part of an extensive gene expression study [4], a large number of gene expression images of sagittal sections of the brain of 7-day old (P7) mice were acquired by researchers at Baylor College of Medicine, Houston, Texas. The study of P7 mouse brains is of special interest partly because the cortical map of the brain is actively forming at this age, and also because previous research focused on either the mouse embryo or adults [7]. The brains, measuring about 60mm x 80mm x 120mm, were extracted from the C57BL6 mouse strain and placed into a freezing chamber. The brain was visually aligned to keep the brain axes parallel to the walls of the chamber. The brains were sliced sagitally, from the left side to the midline, into $20\mu m$ thick sections inside a cryostat at a temperature of $-20^{\circ}C$. Generally, sections from a single brain were distributed into 7 different sets. Each set was then assigned to a specific gene probe. This would result in approximately 24 sections per gene, with a spacing of $140\mu m$ between sections of a gene. *In situ* hybridization was performed on each tissue section to reveal the expression of a specific gene before the sections were imaged with a microscope at 50x magnification, resulting in images with a resolution of $3.3\mu m$ per pixel. The resulting images, with a dimension of about 3000x2000 pixels, were automatically assembled from smaller images since the entire tissue was too big to be imaged at once. This automatic mosaicing process is performed by software associated with the microscope and is described in detail in [4].

ISH is a widely used method for visualizing cellular gene expression in tissue sections. ISH in this context is the binding of the mRNA of interest with its synthesized complementary RNA sequence, also known as the anti-sense riboprobe, inside the tissue (*in situ*). Riboprobes are labeled to make the hybridization results visible. Labeling with radioactive materials requires a long exposure time and sometimes leads to inaccurate results since the radioactive materials do not penetrate the cells [4]. On the other hand, by labeling the riboprobes with digoxygenin antibodies (which also make the dye precipitate stay inside the cell boundaries), visualization is possible within a few hours. ISH results in blue/purple dye precipitates which are clearly visible in the tissue. ISH is relatively simple and is easily automated for high throughput processing. High throughput ISH [2] was developed to automate the ISH process and for processing a large number of genes.

Gene expression images pose a number of unique challenges for segmentation and image registration. Each individual mouse brain is inherently slightly different from every other one in structure, although the basic anatomical features are the same. This results in a large variation in shape. Since each gene is expressed differently from every other one, the expression images vary substantially in appearance. There are also many region boundaries where no edge or intensity variation can be visually observed. In spite of the extreme precision employed in carrying out the HITISH process, sectioning induces local distortion upon tissue slices and artifacts (e.g., air bubbles, missing parts, and smears) are sometimes introduced during the data acquisition process (Fig. 1). The main differences between HITISH and other histological images are:

i) The primary signal in HITISH is the gene expression, which can vary drastically from gene to gene; and
ii) Non-expressing cells are a background signal with HITISH, while with regular histological staining (e.g., Nissl), every cell is highlighted as signal, making further processing easier [7].

A survey of the literature reveals several attempts to register histological sections of the mouse brain. Ali and Cohen [8] matched coronal sections of the brain to an atlas using geometric curve invariants and B-splines. However, the B-spline parameterizations were employed only for the outer boundaries and no specific effort was made to match internal regions. Efforts to reconstruct a 3D brain from 2D histological sections [9], [10] have typically used a uniform stain for all the cross sections.

Accurate segmentation of gene expression images into distinct anatomical regions and subregions is a major step towards efficient characterization of gene expression patterns. This will answer questions about *where* specific genes are expressed in the brain. It will also facilitate the comparison of the expression patterns of individual genes. To compare gene expression patterns across images, these images must first be placed in a common coordinate system. Thus, the need for a brain *atlas*. An atlas partitions an image into disjointed anatomical regions labeled accordingly. Deformable atlases are a fundamental data structure in bio-medical modeling. Warping an atlas to fit an instance of an anatomical shape allows data associated with the atlas to be mapped on the shape in question. Given several sets of gene expression data, the warped atlas provides a common domain on which these

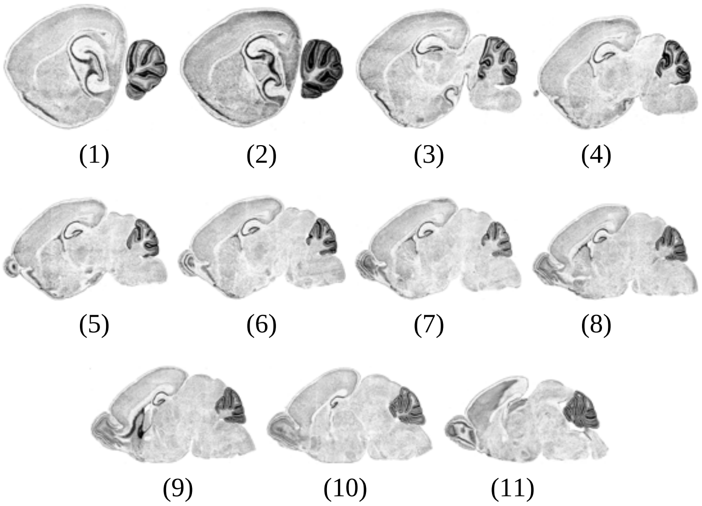


Fig. 2. Eleven standard sagittal cross sections of the mouse brain (Nissl-stained) corresponding to the 11 Valverde atlases.

datasets may be compared.

Some previous mouse brain digitization efforts used atlases that were either pixelated or polygonal. A pixelated atlas, such as the Harvard mouse atlas [11], is a reference image in which each pixel (or voxel) is associated with an anatomical region. A target image is then warped onto the reference image using a tractable class of tensor product deformations [12], and regions in the reference image are mapped onto the target image. Although pixelated atlases are easy to construct, the region boundaries exhibit a 'stair-step' character. On the other hand, a polygonal atlas such as that used by the Mouse Atlas Project at UCLA [13] is constructed from a network of polygonal curves partitioning a reference image into anatomical regions. For a given stack of images, each 3D region is constructed into a separate polygonal representation by connecting the region's 2D contours to form a 3D contour. As in the case of pixelated atlases, the boundaries of the polyhedral atlas are not smooth representations. The most significant defect is that the separate regions generally overlap or have gaps. Atlases based on subdivision surfaces avoid these drawbacks.

Modeling with subdivision surfaces has many advantages including smooth parametrization, few control points to describe dense surfaces, ability to model arbitrary topology, automatic level of detail, numerical stability of solution, and ease of implementation [14]. Although there has been a great deal of work in model-based image segmentation, little work [15], [16], [17] has been reported in the literature on the use of subdivision-based models in biomedical applications. In this paper[1], we present a framework to deform a subdivision-based statistical atlas to tissue sections revealing gene expression patterns. In a closely related work, Ng *et al.* [6] registered a stack of 2D gene expression sections to a reference 3D mouse brain atlas, relying mainly on the outer boundaries. The delineation of the internal regions was 'gross' [6] and no quantitative assessment was provided.

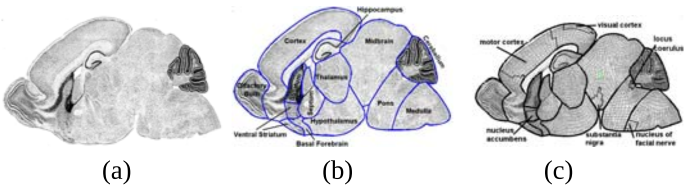


Fig. 3. (a) Image assigned to cross section 9 atlas. (b) Anatomical regions and (c) typical subregions in this cross section.

III. Hybrid Anatomical Atlas

In our previous work [20], subdivision meshes were used to construct 11 atlases corresponding to the 11 standard sections in Valverde's [21] atlas of the postnatal mouse brain. Eleven standard sections were previously determined as the minimum number of sections to annotate while sampling all major nuclei of the brain, which are generally larger than $500\mu m$ in diameter [7]. From each set of 24 tissue sections representing the expression of a specific gene, the 11 tissue sections that best matched the 11 standard sections were manually assigned by neuroanatomists based on the presence of specific anatomical features. For example, the thalamus first appears in section 2. There is enough midbrain to connect all of the brainstem and cerebellum with the thalamus in section 4. The first ventral appearance of the hippocampus is in section 6 and the septum and striatum show in approximately equal proportions in section 9. The 11 standard cross sections are illustrated in Fig. 2. The atlas was labeled with 15 major anatomical regions: cerebellum, cortex, basal forebrain, hippocampus, hypothalamus, medulla, midbrain, olfactory bulb, pons, septum, striatum, thalamus, ventral striatum, and amygdala. Fig. 3(b) depicts the regions present in atlas cross section 9 (Fig. 3(a)). Arbitrary subregions such as the visual cortex, motor cortex, nucleus accumbens, substantia nigra, and locus coerulus can easily be represented in the mesh (Fig. 3(c)). Note that the underlying boundary of any region or subregion is a set of smooth curves that can be parametrically obtained from vertices of the mesh of that region/subregion.

To fit a standard atlas to a given section, the atlas was deformed using global affine transformation to account for rotation and translation of the images, as well as variation in scale. This was followed by local deformation of the image boundary based on iterated least-squares to account for shape variation. However, the accuracy of the local fitting, and interior coordinate system resulting from the segmentation, is limited by its reliance only on the detection of the external tissue boundary. Thus, further deformation of the internal regions of the atlas must still be performed.

In this work, we have augmented the atlas with statistics of the shape of the regions, locations and appearance of selected anatomical landmarks, and models of textural variation at anatomical region boundaries. Our atlas $A = \{A_1, A_2, \ldots, A_{11}\}$ consists of 11 hybrid atlases, where each atlas A_i corresponds to a standard cross section in the sagittal series of the Valverde reference atlas. Specifically, the atlas for each of the 11 standard cross sections is associated with a

[1] Part of this work appeared in [18] and [19].

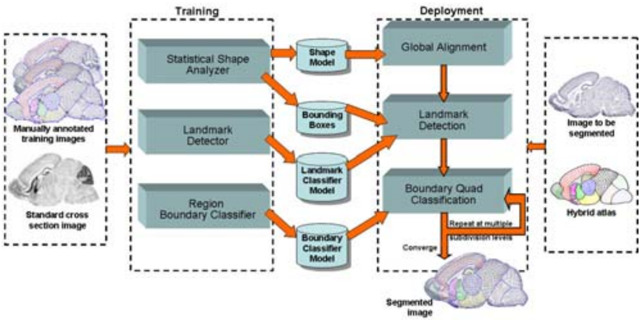

Training
Deployment
Manually annotated training images
Standard cross section image
Statistical Shape Analyzer
Landmark Detector
Region Boundary Classifier
Shape Model
Bounding Boxes
Landmark Classifier Model
Boundary Classifier Model
Global Alignment
Landmark Detection
Boundary Quad Classification
Repeat at multiple subdivision levels
Converge
Segmented image
Image to be segmented
Hybrid atlas

Fig. 4. Overview of the proposed segmentation framework.

triplet such that:

$$A_i = \{S_i, L_i, B_i\}\ ,\ i \in \{1, \ldots, 11\} \tag{1}$$

where S_i represents the shape of the subdivision mesh that captures the geometry and topology of the atlas, L_i represents selected anatomical landmarks associated with the mesh, and B_i represents the appearance on the boundaries of anatomical regions. Our proposed learning-based segmentation framework consists of a training and a deployment stage (Fig. 4). Several parameters of the atlas are learned by *training*. The learned parameters are then used to segment new images at the *deployment* stage. Hence, the use of the term *learning-based* to describe our segmentation framework. Training is performed on several mouse brain gene expression images on which the hybrid atlas was manually fitted by neuroanatomists. During deployment, this atlas is automatically fitted to new gene expression images in order to segment them into anatomical regions. The elements of the atlas are explained in this section while the training and deployment stages are discussed in the Sections IV and V.

A. Statistical Shape: Subdivision-based Mesh

We propose the use of an anatomical atlas for the mouse brain based on Catmull-Clark subdivision surfaces [22]. Subdivision is a fractal-like process of generating a smooth geometry from a coarse shape [14]. Starting from an initial mesh M^0 which was manually constructed to model a standard tissue cross section, subdivision generates a sequence of refined meshes $(M^1, M^2, ..., M^k)$ with increasing smoothness. In the following, we briefly describe how the atlas is constructed using subdivision. More details can be found in [20].

Catmull-Clark subdivision is designed for 2D meshes that consist of quadrilaterals (or 'quads' for short). Each subdivision step can be broken down into two simple transformations: bi-linear subdivision, followed by centroid averaging. Under bi-linear subdivision, each quad is split into four sub-quads with new vertices placed at the midpoints of old edges and at the centroids of old faces. Under centroid averaging, each vertex in the mesh is re-positioned to the centroid of the centroids of those quads that contain the vertex. To model the partitioning of the mesh into sub-meshes, a subset of edges and vertices in the initial mesh M^0 are flagged as *crease edges* and *crease vertices*. Crease edges are used to partition the mesh into submeshes representing the anatomical regions. Crease vertices are designed not to move during the entire process of subdivision. In the first phase (bi-linear subdivision), each crease edge is linearly subdivided into two crease edges. Next, during centroid averaging, each vertex on a crease edge is repositioned at the centroid of the midpoints of those crease edges that contain the vertex. The positions of crease vertices are left unchanged.

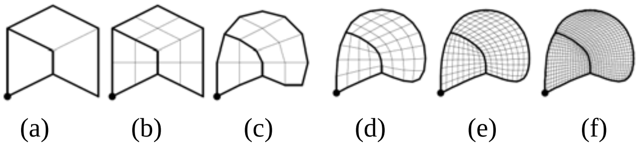


Fig. 5. Illustration of subdivision process [20]: (a) initial mesh; (b) after bi-linear subdivision; (c) after centroid averaging; and after subdivision (d) two, (e) three, and (f) four times.

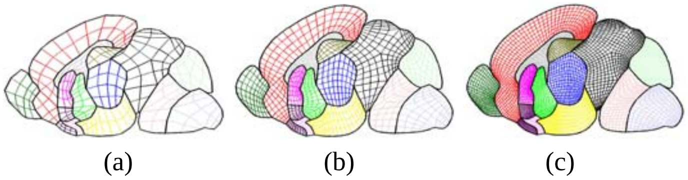


Fig. 6. The mesh of the hybrid atlas at subdivision levels (a) 1, (b) 2, and (b) 3. The term 'subdivision level' denotes how many times the initial mesh has been subdivided.

The subdivision process is illustrated in Fig. 5. Starting with an initial quadrilateral mesh M^0 containing crease edges and vertices (Fig. 5(a)), one round of subdivision consists of bi-linear subdivision (Fig. 5(b)) and centroid averaging (Fig. 5(c)). As more subdivisions are performed (Figs. 5(d-f)), the quadrilaterals are refined and the crease edges approximate smoother curves. The limit of this subdivision process is a smooth mesh M^∞ partitioned by a network of cubic B-spline curves that interpolate crease vertices of M^0 and approximate the curve network formed by the crease edges in M^0.

Using subdivision, the mouse brain atlas is represented as a coarse quadrilateral mesh partitioned by crease edges into sub-meshes corresponding to each major anatomical region of the mouse brain. The initial mesh M^0 is constructed by experts in such a way that repeated subdivisions of M^0 will result in accurate and smooth partitioning of the standard image into anatomical regions. Fig. 6(a-c) depict the brain atlas subdivided once, twice and three times. Each quad of the initial atlas is tagged by its corresponding anatomical region (indicated by colors in Fig. 6), and this information is propagated to subdivided quads during bi-linear subdivision.

The shape term, S, of the hybrid atlas defines the geometry and topology as well as the statistical modes of variation of the subdivision mesh that will be fitted to each image. The geometry is a collection of the coordinates of the vertices of the mesh at a given subdivision level while the topology denotes the relationships between the vertices to form anatomical regions. The geometry of a mesh with n such vertices is represented as:

$$\mathbf{x}^k = [x_1^k, x_2^k, \ldots, x_n^k, y_1^k, y_2^k, \ldots, y_n^k]^T, \tag{2}$$

at subdivision level k, where $\mathbf{v}_j^k = [x_j^k, y_j^k]$ are the coordinates of vertex j. In our hybrid atlas, the shape term for each cross section atlas is a representation of the shape at multiple subdivision levels such that:

$$S_i = \{S_i^k\}, k \in \{1, 2, 3\}, \tag{3}$$

where the shape at each subdivision level, k, is expressed as a linear combination of the mean shape $\bar{\mathbf{x}}^k$ and the shape vectors $\mathbf{\Phi}^k$ at that level, i.e.,

$$S_i^k = \bar{\mathbf{x}}^k + \mathbf{\Phi}^k \mathbf{b}^k. \tag{4}$$

The mean shape is obtained for N manually fitted meshes in a training set as:

$$\bar{\mathbf{x}}^k = \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}^k, \tag{5}$$

while the shape vectors, $\mathbf{\Phi}^k = (\mathbf{\Phi}_1^k, \mathbf{\Phi}_2^k, \ldots, \mathbf{\Phi}_t^k)$, are learned at the training stage. They are the eigenvectors corresponding to the eigenvalues $\lambda^k = (\lambda_1^k, \lambda_2^k, \ldots, \lambda_t^k)$, which represent the t principal modes of variation that account for a given percentage of the variation of the shape. The shape parameters $\mathbf{b}^k = [b_1^k, b_2^k, \ldots, b_t^k]$ control the contribution of each eigenvector in a given shape. Consequently, each shape has a unique set of shape parameters.

Due to the linearity of the subdivision process, the geometry of each subdivided mesh $\mathbf{x}^k$ depends only on the position of the vertices in the coarse mesh $\mathbf{x}^0$. In particular, the position of an individual vertex $\mathbf{v}_j^k$ on the kth subdivision mesh depends only on the positions of the vertices of the coarse mesh $\mathbf{x}^0$ via

$$\mathbf{v}_j^k = \mathbf{u}_j^k \cdot \mathbf{x}^0. \tag{6}$$

The weight vector $\mathbf{u}_j^k$ depends on the choice of particular subdivision scheme and the relative topological position of the vertex $\mathbf{v}_j^k$ in the mesh $\mathbf{x}^k$. In the case of Catmull-Clark subdivision scheme, the linear relation between a vertex at subdivision level k and its neighboring vertices at subdivision level $k-1$ is illustrated in Fig. 7. Multiplying these linear relations at each subdivision level yields the weight vector $\mathbf{u}_j^k$, which expresses the linear relation between a vertex at subdivision level k and vertices at level 0. Figure 7 depicts the Catmull-Clark subdivision mask for positioning a vertex (the dark dot) on a fine mesh $\mathbf{x}^k$ with respect to the neighboring vertices of a coarse mesh $\mathbf{x}^{k-1}$. Given a particular choice of subdivision scheme, the weight vectors $\mathbf{u}_j^k$ can be computed using subdivision. If we initialize the weight vectors $\mathbf{u}_j^0$ on the coarsest mesh to have value 1 at the ith vertex and 0 elsewhere, applying the subdivision process to these vectors k times yields the desired weight vectors $\mathbf{u}_j^k$. Warren and Weimer [14] provide detailed information on specific masks to use at crease vertices and in other special cases.

B. Landmarks

The second component, L_i, of the hybrid atlas includes a set of anatomical and geometrical landmarks in the atlas of each cross section i such that:

$$L_i = \{{}^{B}L_i, {}^{I}L_i\}. \tag{7}$$

The component L_i captures statistical information about the location and appearance of selected landmarks in the image. These landmarks are locations in the image that have a relatively consistent shape and appearance irrespective of the gene being expressed. The landmarks are of two types: boundary and internal.

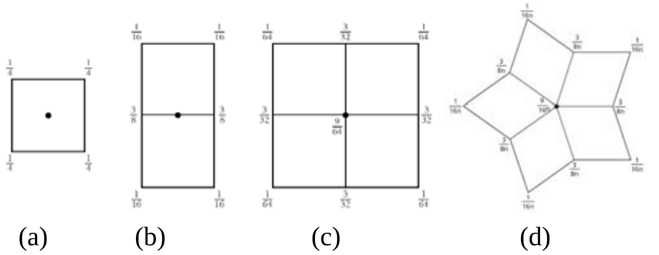


Fig. 7. Examples of the use of matrix $\mathbf{u}_j^k$ to determine the position of a vertex (the dark dot) on a fine mesh $\mathbf{x}^k$ with respect to the neighboring vertices of a coarse mesh $\mathbf{x}^{k-1}$ [14]. For the generalized case in (d), n is the number of quads sharing the center vertex. Note that $n = 4$ in (c).

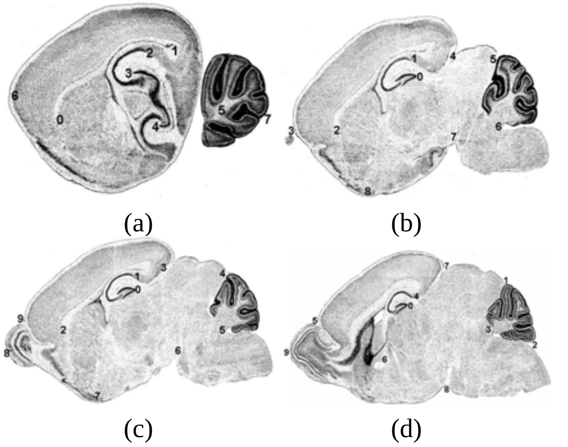


Fig. 8. Selected landmarks in tissue sections assigned to atlas cross sections (a) 2, (b) 4, (c) 6, and (d) 9 respectively.

Boundary landmarks, ${}^{B}L_i$, are identified as points of extreme curvature along the outer boundary of the brain section. These are usually points where two regions meet on the outer boundary. On the other hand, internal landmarks, ${}^{I}L_i$, are identified as parts of anatomical organs in the image that have some characteristics which make them distinguishable, fairly consistently, irrespective of the particular gene expression. Such characteristics include extremely high cell density, such that most genes are expressed in the organ to some extent (e.g., in the dentate gyrus and parts of the cerebellum); or overwhelming presence of fiber tracts such that few or no genes are expressed in them (e.g., in the anterior commissure). Examples of both boundary and internal landmark types for different cross sections are depicted in Fig. 8.

Each internal landmark is further associated with the following six elements:

(i) A vertex, $\mathbf{v}_j$, in the atlas at subdivision level 3 (an appropriate level to accurately represent the selected landmarks). Each $\mathbf{v}_j$ in $\mathbf{x}^3$ (mesh at subdivision level 3) is obtained from vertices in $\mathbf{x}^0$ (mesh at subdivision level 0) as described in Eq. 6. During deployment, effort is made to match the detected location of a landmark in the image with its corresponding vertex in the atlas.

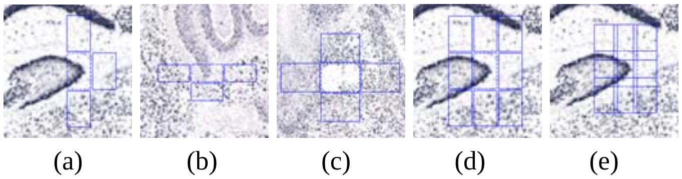


Fig. 9. (a-d) Some possible arrangements of templates on landmarks. (e) All 8 neighbors with reduced inter-template distance.

(ii) The location $\boldsymbol{\mu}_j$, which is the average location of this landmark point in a set of representative training images. The average is computed after all images in the training set are aligned to the standard image for the section. The alignment is described in Section IV-A.

(iii) The standard deviation, $\boldsymbol{\sigma}_j$, of the location of the landmark in all training images, which is used in conjunction with $\boldsymbol{\mu}_j$ to define a rectangular bounding box within which to search for the landmark in new images. The bounding box is centered on the mean and extends by $\pm D\boldsymbol{\sigma}_j$ in x and y directions, where D is chosen to capture up to 99% of the variation in landmark locations.

(iv) A set of template parameters, $\mathcal{T}_j$ that define a rectangular template around the landmark in which features are to be extracted. The size of the template (specified by height and width) for each landmark is part of the template parameters to be decided before training. The arrangement of the template in neighboring areas to extract features for the 'out-class' (Figs. 9(a-d)) and the distance between the landmark and neighbor templates (Fig. 9(e)) are also specified in $\mathcal{T}_j$.

(v) The optimal set of texture features, ${}^{L}F_j$, that are specifically suited for identifying the landmark and are learned from training images using methods described in Section IV.

(vi) A set of classifier parameters, ${}^{L}P_j$, also learned from training, which define a classifier that can successfully distinguish a landmark within its neighborhood based on ${}^{L}F_j$.

Thus, each internal landmark j at section i is modeled as:

$$ {}^{I}L_i^j = \{\mathbf{v}_j, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, \mathcal{T}_j, {}^{L}F_j, {}^{L}P_j\}. \tag{8} $$

On the other hand, boundary landmarks are modeled as

$$ {}^{B}L_i^j = \{\mathbf{v}_j, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, c_j\}, \tag{9} $$

where $\mathbf{v}_j$, $\boldsymbol{\mu}_j$, and $\boldsymbol{\sigma}_j$ are defined similarly as for the internal landmarks. Additionally, each boundary landmark is associated with the type of curvature, c_j, which can be convex (positive curvature values) or concave (negative values) since boundary landmarks are points on the tissue boundary with the largest curvatures within their respective bounding boxes. Note that $\mathbf{v}_j$, $\mathcal{T}_j$ and c_j are user-specified. The other parameters namely, $\boldsymbol{\mu}_j$, $\boldsymbol{\sigma}_j$, ${}^{L}F_j$, and ${}^{L}P_j$, are learned from training.

C. Region Boundary

To ensure accurate delineation of the anatomical region boundaries, texture features in the quads on region boundaries are included in the model and they are used to guide the segmentation process.

> **Definition 1:** *A boundary quad is a quad that has at least one side belonging to more than one anatomical region*.

A corollary of this definition is that boundary quads are those that are not completely enclosed inside a single anatomical region. All boundary quads have at least one side being a crease edge since the anatomical regions are partitioned with crease edges. Learning the appearance variation between adjacent boundary quads is, in effect, learning to delineate the anatomical boundaries. Contiguous quads on the boundary of two anatomical regions are grouped into *segments*.

> **Definition 2:** *A boundary segment is the maximum length of a curve on an anatomical boundary that has quads from no more than two anatomical regions.*

For example, the thalamus is bounded by five segments: thalamus/hippocampus, thalamus/midbrain, thalamus/hypothalamus, thalamus/septum, and thalamus/fibers (Fig. 10). Note that a boundary segment that contains quads from only one region is on the external boundary of the mesh and is adjacent to the image background. By separating the region boundaries into segments, optimal features for each segment can be chosen, since no set of features will be equally optimal for all region boundaries.

This arrangement makes it possible to treat the search for region boundaries as a binary classification problem. During deployment, the objective then becomes finding a boundary such that quads on one side are correctly labeled as belonging to one class, say '$+1$', and quads on the opposite side as another, say '-1'. For each segment j at subdivision level k, let Q_j^k be the set of all quads on the segment, labeled as '$+1$' or '-1' depending on the anatomical region they belong to; let ${}^{B}F_j^k$ be the set of optimal features for identifying the quads, and ${}^{B}P_j^k$ the set of classifier parameters to distinguish between quads on opposite sides of the boundary segment. For each segment,

$$ B_j^k = \{Q_j^k, {}^{B}F_j^k, {}^{B}P_j^k\} \quad j \in \{1, \ldots, s\}, \tag{10} $$

and

$$ B^k = \{B_1^k, B_2^k, \ \ldots \ , B_s^k\}, \tag{11} $$

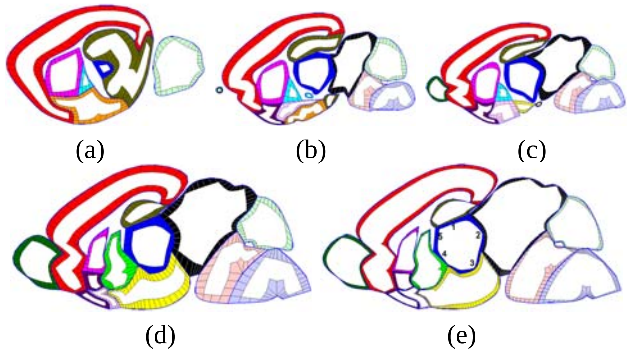
for all s segments in the mesh at subdivision level k. Note that ${}^{B}F_j^k$ and ${}^{B}P_j^k$ are to be learned from training.

IV. Atlas Training

During training, the parameters $\boldsymbol{\mu}_j$, $\boldsymbol{\sigma}_j$, ${}^{L}F_j$, ${}^{L}P_j$, ${}^{B}F_j^k$, and ${}^{B}P_j^k$ are obtained through the following processes:

A. Training: Statistical Shape

In building the model, we wish to obtain the mean shape of the mesh along with the principal modes of variation of the vertices as a way of constraining the deformation of the shape. This is repeated for different subdivision levels (up to level 3 in this work). To obtain the mean shape, all the images in the training set are aligned to the standard image for the cross



(a) (b) (c)

(d) (e)

Fig. 10. Boundary quads at subdivision level 2 for atlas cross sections (a) 2, (b) 4, (c) 6, and (d) 9. (e) Boundary quads at subdivision level 3 for atlas cross section 9. Boundary segments for the thalamus are numbered 1-5 in (e).

section. The global alignment, adapted from Ju *et al.* [20], is conducted in five steps as follows:

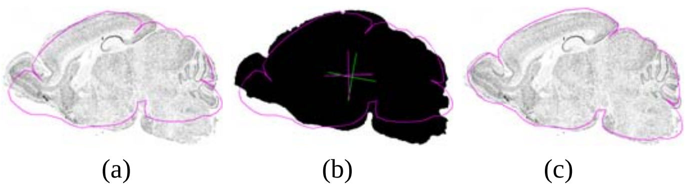
(i) Histogram thresholding: The standard image for the cross section is converted to grayscale (0-255) and the histogram obtained. An intensity threshold value is dynamically selected such that about 60% of the image would be labeled as background. This threshold value was always between 251 and 254 in all the images tested, since the background is almost white (intensity value 255). The percentage 60% was experimentally determined.

(ii) Background segmentation: The tissue section is segmented from the background in the thresholded image using a *flood filling* method from the top left corner. In flood filling, the first pixel is taken to be part of the background and all the pixels that are connected to it (in a recursive region growing fashion) and having similar intensity values are also considered background. This successfully segments the tissue, but also leaves artifacts on the background and on the tissue boundaries.

(iii) Artifacts removal: After smoothing with a Gaussian filter, artifacts are removed by finding the largest component in the segmented image and all other components are considered noise. This is possible since there is only one piece of tissue in most sections. In atlas cross sections 1 and 2 where the cerebellum is separated from the brain, and in atlas cross section 4 where the olfactory bulb is separated, we selected the largest two components.

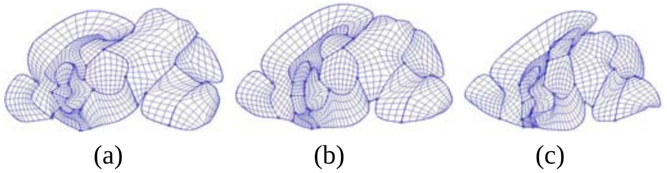
(iv) Computation of principal axes: The coordinates of the centroid $\mathbf{o}$ of the segmented image is computed as $\frac{1}{m}\sum \mathbf{a}_j$ where $\mathbf{a}_j$ is a column vector of the coordinates of each pixel j in the segmented tissue and m ia the number of pixels in the segmented tissue. Using PCA, the two orthogonal eigenvectors of the 2x2 covariance matrix $\frac{1}{m-1}\sum_{j=1}^{m}(\mathbf{a}_j - \mathbf{o})(\mathbf{a}_j - \mathbf{o})^T$ represent the first and second principal variations of data points, and thus give the general orientation of the image.

(v) Affine alignment: Steps (i) - (iv) are similarly conducted for the image to be aligned to the standard image. To obtain the alignment parameters, the required translation is computed as the difference between the two centroids; the difference between the orientation (from the principal axes) indicate the angle of rotation; and the scaling parameters are obtained from the ratio of the two pairs of eigenvalues (Fig. 11).



(a) (b) (c)

Fig. 11. Alignment of the principal axes of the atlas and image in the *CB-FAT2T1* expression image. (a) Outer boundary of the atlas before alignment. (b) Principal axes of atlas and segmented image. (c) Outer boundary of atlas after alignment.



(a) (b) (c)

Fig. 12. The shape of the quads changes as the model weights for the first four principal components are varied and assume the values (a) -2λ, (b) 0, and (c) $+2\lambda$, respectively.

The mean shape is obtained using Eq. 5. In order to obtain the shape vectors, $\mathbf{\Phi}^k$, we first obtain the covariance matrix for the N shapes in the training set:

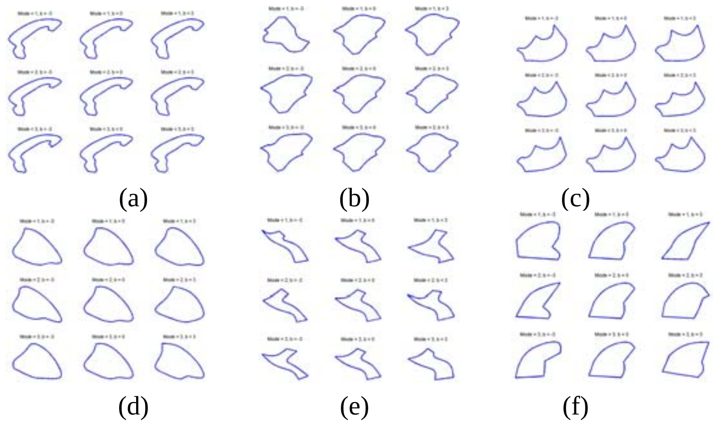
$$\frac{1}{N-1}\sum_{j=1}^{N}(\mathbf{x}_j^k - \bar{\mathbf{x}}^k)(\mathbf{x}_j^k - \bar{\mathbf{x}}^k)^T. \qquad (12)$$

Using PCA, the eigenvectors of the covariance matrix are computed along with their corresponding eigenvalues. The eigenvectors (or shape vectors in our case) which correspond to the principal modes of variation of the shape are then sorted in decreasing order of their eigenvalues. The largest eigenvectors corresponding to the largest t eigenvalues accounting for at least 95% variation in the shape are constituted into $\mathbf{\Phi}^k$. The modes of variation of all the vertices of the atlas are depicted in Fig. 12 while the variation in individual anatomical regions is depicted in Fig. 13.

B. Training: Landmarks Detection

For each landmark j, the mean position, $\boldsymbol{\mu}_j$, and the standard deviation, $\boldsymbol{\sigma}_j$, of the landmark were obtained after performing an affine alignment of the training images. The landmark detector was trained to identify each internal landmark within its neighborhood in five steps: template placement, feature extraction, feature normalization, optimal feature selection, and classifier parameters computation.

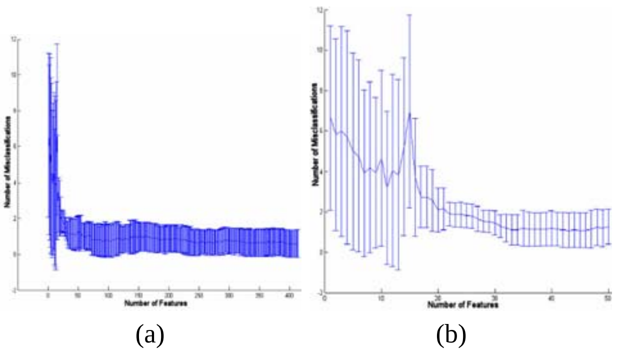
1) Template Placement: To extract features around a landmark point, a manually defined rectangular template was chosen to overlay distinguishing features of the landmark while excluding neighboring areas. After examining the area around the landmarks in several images, a template with a dimension of 21x11 pixels was used for the landmarks (except



(a) (b) (c)

(d) (e) (f)

Fig. 13. Principal modes of variation of the shape of the (a) cortex, (b) midbrain, (c) hypothalamus, (d) cerebellum, (e) basal forebrain, and (f) striatum.



(a) (b)

Fig. 14. Plot of the number of features in a *forward selection* process against the average classification error with error bars representing standard deviation in a 4-fold cross validation. (a) Use of 414 features for landmark 0. (b) A closer view of the first 50 features in (a).

the anterior commissure, where a 35x25 pixel rectangle was used). The same template was positioned in locations that are quite close to the landmark and yet are different enough to be considered as typical examples of what the landmark is not (i.e., ‘out-class’). The specific arrangement of the 3-neighborhood ‘out-classes’ was decided experimentally with the objective of maximizing the difference between the ‘in-class’ and ‘out-classes’ while maintaining a high level of similarity between the ‘out-classes’ (Fig. 9).

2) Feature Extraction: Although the model does not constrain the type of features to use, we used features based on the statistics of local histograms [23] in our experiments. The algorithm for extracting the features is as follows:

- Convert the original image to grayscale (I_0) using the standard NTSC RGB to greyscale conversion: 0.299R + 0.587G + 0.114B, and convolve with two Gaussian kernels (size = 5x5 and σ = 1, 2) resulting in two smoothed images (I_1, I_2).
- Convolve the two images with 4 edge detector filters (45°, 90°, 135°, and 180°). This results in 8 filtered images.
- Divide the rectangular region around the landmark overlaid by the template into overlapping 5x5 windows, two pixels apart. Obtain the local histogram of each subwindow and compute the mean, standard deviation, and skewness.
- Obtain similar features with overlapping 9x9 windows. (The size of the subwindows - 5x5 and 9x9 - were determined experimentally).
- Label the resulting feature vector as ‘+1’ (‘in-class’).
- Use the same template to extract features from the neighbors and label as ‘-1’(‘out-class’).

The resulting features are essentially the first, second, and third moments of an ordered set of subwindows in the template. There were a total of 414 features using a 21x11 template for landmarks in atlas cross section 9. The list of features is prohibitively long to include here, but we will be happy to provide it to interested researchers.

3) Feature Normalization: Many feature normalization techniques, including linear scaling to unit range, linear scaling to unit variance, transformation to uniform random variable, and rank normalization have been used in texture analysis problems [24]. We selected linear scaling to unit range due to its simplicity, and the need to preserve the scaling parameters for use during deployment. In this approach, the feature mean is subtracted from each feature value and the result is divided by the feature standard deviation, to prevent the domination of some features by others.

4) Optimal Feature Set Selection: It is expected that not all the features extracted will contribute equally to the classification. The features were reduced to an ‘optimal’ set using *forward selection* [25] which consists essentially of ranking the features in order of relevance to the classification problem, and evaluating the performance of the classifier using an increasing number of features. We computed the relevance of each feature using the Information Gain [25] metric which has been found to be very reliable in previous applications [26]. However, the features need to be discretized; Fayyad and Irani’s minimum description length algorithm [27] was used for this. The features were then sorted in order of relevance and each feature was included one-at-a-time in a feature set, ${}^{L}F_j$. The average error of classifying with the feature set was obtained in a 4-fold cross-validation [28] in which the data is divided into four subsets and one set is alternately used for validation while the other three are used for training. The average error over the four attempts is taken as the classification error. As the number of features in the set increases, the average error generally decreases until relative stability is attained. The smallest set of features, ${}^{L}F_j$, with which the error becomes stable and yet lower than a given threshold is selected as the optimal feature set. Figure 14 illustrates a typical situation when the error becomes stable after including about 40 features. Most of the features in the optimal set are the mean of the histogram distribution in the 5x5 windows in the 45° and 135° filtered images. In atlas cross section 9, the number of features in the optimal set was 40, 75, and 65 in internal landmarks 0, 4, and 3 respectively.

5) Classifier Parameters Computation: For the classification task, we considered the use of k-Nearest Neighbor and various Linear Discriminant Analysis [28] classifiers. A Support Vector Machine (SVM) [29] classifier was selected due to its ability to give reasonable decision boundaries even with sparse data sets. Given a set of h training instances, where each instance $\mathbf{z}_i$ consists of a set of features ${}^{L}F_j$ and a class label y_i, the goal of SVM is to map the training examples into a higher dimensional hyperspace and then produce an optimal separating hyperplane in that space. This is typically formulated as an optimization problem [30]:

$$\begin{aligned} \min_{\boldsymbol{w},b,\xi} \quad & \tfrac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \;+\; C\sum_{i=1}^{h}\xi_i \\ \text{subject to}: \quad & y_i(\boldsymbol{w}^T\phi(\mathbf{z}_i)+b) \geq 1-\xi_i, \\ & \xi_i \geq 0, i=1,\dots\, h, \end{aligned} \tag{13}$$

where ϕ is the mapping function or kernel used to map the data into higher dimensional space; $\boldsymbol{w}$ is a vector normal to the hyperplane that represents the decision boundary; the constant $C > 0$ is the penalty parameter for the hyperplane separation error; and ξ is a slack variable used to penalize the objective function. The inner product $\phi(\mathbf{z}_i)^T\phi(\mathbf{z}_j)$ is usually replaced by a kernel $K(\mathbf{z}_i, \mathbf{z}_j)$ to minimize computational cost. In our framework, we utilized the Radial Basis Function (RBF) kernel [31] where:

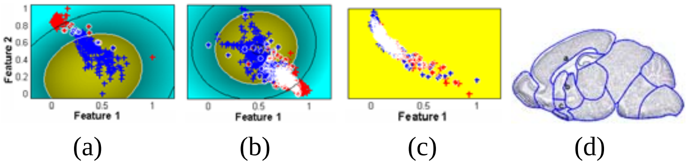
$$K(\mathbf{z}_i, \mathbf{z}_j) = exp(-\gamma||\mathbf{z}_i - \mathbf{z}_j||^2), \gamma > 0. \tag{14}$$

Optimal values for C (hyperplane separation error parameter) and γ (RBF kernel inverse width parameter) are obtained by cross validation using values of C between 2^{-5} and 2^{25}, and γ between 2^{-15} and 2^{3}, as suggested in Hsu *et al.* [30]. The optimal C and γ along with the SVM support vectors, constitute the SVM model parameters, ${}^{L}P_j$, for each landmark. Given a set of model parameters and a data instance to classify, the result of SVM is a *decision value*. When the decision value is positive, the data instance is assigned to the 'in-class' (label: +1) and a negative value implies an 'out-class' instance (label: -1). The magnitude of the decision value was consistently higher in landmarks that can be easily identified in their neighborhood compared to landmarks that are not clearly distinguishable, lending it to be used as a weight factor.

C. Training: Region Boundary Classifier

The third element of the hybrid atlas is the region boundary classifier which captures information about texture feature variation at anatomical region boundaries. It was observed that the cell density pattern in the cerebellum varies from that of neighboring regions in most images. Similarly, texture variation can be observed along the boundaries of the cortex, septum, and thalamus. In addition, the non-expressing cellular background signal forms part of the cell density texture as does the gene expression signal. This variation in the texture patterns of anatomical regions is captured in the boundary quads model at multiple subdivision levels. The optimal feature set, ${}^{B}F_j^k$, and classifier parameters, ${}^{B}P_j^k$, are obtained as follows:

1) Step 1: Feature extraction from quads: Each image is filtered with Laws' texture filters [32] to obtain a bank of 25 texture energy maps. Each Laws' energy map emphasizes to varying degrees the ripple, wave, level, spot, and edge features of the images. In each of these texture energy maps, the pixels belonging to each boundary quad are obtained, and the first, second, and third moments of the distribution of intensity values in the quad are computed. These are used as texture features for the quad.



(a) (b) (c) (d)

Fig. 15. Illustration of SVM decision boundaries for regions with (a) clearly, (b) fairly, and (c) poorly distinguishable boundaries. Corresponding image boundaries are labeled in (d). Features 1 and 2 are the Laws features with the highest ranking using Information Gain Ratio metric.

2) Steps 2-4: Feature normalization, optimal feature selection, and classifier parameter computation: These steps are essentially the same as described for the landmarks in Section IV-B. Here however, the SVM classifier is trained to distinguish between quads on either side of the boundary segment based on the optimal features. It was observed that some of the region boundaries cannot be accurately distinguished (e.g., between the pons and the medulla). The magnitude of the decision values was significantly lower in such boundaries (Fig. 15). In Fig. 15, features 1 and 2 are the best two features ranked by Information Gain Ratio metric (typically the mean of intensity values in the Laws' LL and SS filtered images).

V. Atlas Deployment

In the deployment stage, the hybrid atlas A_i is deformed to a tissue section in order to segment its anatomical regions in two stages: *global transformation* and *local deformation*. During global transformation, the image is segmented from the background and the principal axes computed as described in Section IV-A. In a similar way, the outer boundary of the mesh is treated as the outer boundary of a segmented image and the area within the boundary treated as pixels inside the image. The centroid and the principal axes are computed and used to determine the rotation, translation and scaling parameters of the atlas to the image.

Local deformation is achieved by solving a single linear system of equations whose variables are the locations of the mesh vertices. The solution to the equations minimizes a quadratic energy function $E^k(\mathbf{x}^0)$ of the form:

$$E^k(\mathbf{x}^0) = \alpha E_U^k(\mathbf{x}^0) + \beta E_L(\mathbf{x}^0) + \delta E_B^k(\mathbf{x}^0) + \tau E_d^k(\mathbf{x}^0) + \epsilon E_S^k(\mathbf{x}^0). \tag{15}$$

The energy term $E_U^k(\mathbf{x}^0)$ is the fitting error of the outer boundary of the mesh to the outer boundary of the image; $E_B^k(\mathbf{x}^0)$ measures the fitting error of the regional boundaries resulting from the classification of the boundary quads; $E_L^K(\mathbf{x}^0)$ measures the error of fit of the anatomical landmarks at a fixed subdivision level 3 in our experiments, since the detection of landmarks is not multi-resolution; $E_d^k(\mathbf{x}^0)$ penalizes non-affine deformation of mesh quads; and $E_S^k(\mathbf{x}^0)$ measures the deviation of the shape from within ±2 standard deviations from the mean shape. The constants α, β, δ, τ, and ϵ are user-defined weights to control the respective contributions of each energy term to the fitting process. Since each minimization term E^k is a quadratic function in the variables $\mathbf{x}^0$, the right-hand side of Eq. 15 can be minimized using a linear solver such as conjugate gradients. Next, we explain the derivation of each of the terms.

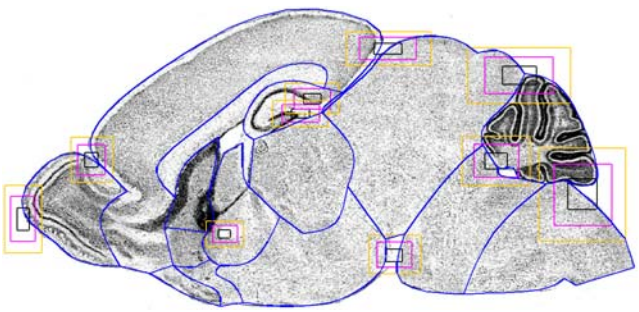


Fig. 16. Extent of the landmark bounding boxes ($\boldsymbol{\mu}_j \pm D\boldsymbol{\sigma}_j$) in cross section 9 image where D = 1 (black colored boxes), 2 (magenta colored boxes), and 3 (orange colored boxes).

A. Outer Boundary Deformation Energy

We have adopted the construction for $E_U^k(\mathbf{x}^0)$ in Ju *et al.* [20] which is a variant of the *iterated closest point* method. Specifically, given points $\mathbf{q}_j$ on the outer boundary of the image after the initial segmentation from background, the fitting error is a quadratic function of the form

$$E_U^k(\mathbf{x}^0) = \sum_j w_j(\mathbf{n}_j \cdot (\mathbf{q}_j - \mathbf{p}_j(\mathbf{x}^k)))^2, \tag{16}$$

where $\mathbf{p}_j(\mathbf{x}^k)$ indicates the vertex in $\mathbf{x}^k$ that is closest to $\mathbf{q}_j$; $\mathbf{n}_j$ is the estimated normal vector at $\mathbf{q}_j$; and the sum runs over all data points $\mathbf{q}_j$. In this model, the deviation of the mesh vertices $\mathbf{p}_j(\mathbf{x}^k)$ from the tangent lines defined by the pairs $(\mathbf{q}_j, \mathbf{n}_j)$ is penalized.

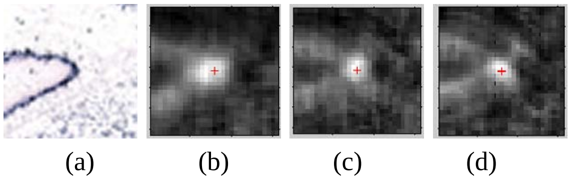
The quality of the fitting term is further improved by replacing simple closest point correspondence (used in pairing $\mathbf{q}_j$ with $\mathbf{p}_j$) with one that favors nearby points with aligned normals. For each data point $\mathbf{q}_j$, the closest vertex in $\mathbf{x}^k$ whose unit normal lies within some small cone of the normal $\mathbf{n}_j$ is computed. In our implementation, this search was restricted to vertices of $\mathbf{x}^k$ that lie within 5% of the image width of $\mathbf{q}_j$ and whose normals differ from $\mathbf{n}_j$ by at most $45°$. The weight term w_j in Eq. 16 is simply the cosine of twice the angular difference between $\mathbf{n}_j$ and the normal for the chosen $\mathbf{p}_j(\mathbf{x}^k)$.

B. Landmark-guided Deformation Energy

For each landmark j, a bounding box is obtained from the mean, $\boldsymbol{\mu}_j$, and standard deviation, $\boldsymbol{\sigma}_j$, of the landmark position in training images. The bounding box is centered at $\boldsymbol{\mu}_j$ and extends to $\pm D\boldsymbol{\sigma}_j$. The multiplier D was chosen as 3 to account for about 99% of the variation of the location of the landmarks (Fig. 16). Since $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ are obtained relative to the standard image and atlas, the bounding box is further scaled to match the image using the affine parameters obtained from the global alignment. A search for each landmark is then conducted within the image area overlaid by this bounding box. Either of two algorithms is followed depending on whether the landmark is internal or on the boundary.

1) Internal landmarks detection: Each internal landmark is detected in three steps:

(i) Feature extraction. The optimal set of features determined at the training stage, ${}^{L}F_j$, are extracted in the area enclosed by the bounding box.



(a) (b) (c) (d)

Fig. 17. SVM probability estimates for pixels in a bounding box around landmark 0 (a) a clear global maximum can be seen around expected ground truth (b-d).

(ii) Feature classification. The SVM classifier model obtained at the training stage is used to classify the extracted features for all the pixels.

(iii) Score computation. In our experiments, many pixels, especially those around the expected landmark point were assigned a label of '$+1$' ('in-class') by the classifier. To determine the best candidate for the landmark point, we used the SVM decision value of the pixel and its immediate neighbors. This is possible because the decision values were found to have a clear global maximum around the expected ground truth in all the images tested (Fig. 17).

While all the boundary landmarks were used for atlas deformation, only internal landmarks that have SVM decision values greater than a given threshold (0.5 in our experiments) at the detected point were used. This excludes some landmarks from the fitting process in images where those landmarks are not effectively detected (e.g., images with no gene expression around the landmarks).

2) Boundary landmarks detection: We applied the border tracing algorithm described by Sonka [33] on the segmented binary image. The resulting boundary is further smoothed using a *moving average* filter to reduce the effect of noise resulting from the flood-filling approach used for segmentation. The coordinates of each point $\mathbf{x}_j = (x_j, y_j)$ on the boundary are replaced by the average position of the point and a number of points before and after it. The number of points was chosen as 16 in our experiments. In Fig. 18, some typical border segments before and after smoothing are illustrated. The curvature c_j at each point on the edge is computed as

$$c_j = \frac{\dot{x}_j \ddot{y}_j - \dot{y}_j \ddot{x}_j}{(\dot{x}_j^2 + \dot{y}_j^2)^{3/2}}, \tag{17}$$

where $(\dot{x}_j, \dot{y}_j) = 1/2[(x_{j+1}, y_{j+1}) - (x_{j-1}, y_{j-1})]$ is an approximation of the first derivative and $(\ddot{x}_j, \ddot{y}_j) = 1/2[(\dot{x}_{j+1}, \dot{y}_{j+1}) - (\dot{x}_{j-1}, \dot{y}_{j-1})]$ approximates the second derivative. For all the sampled edge points within each bounding box, the point of maximum convex or concave curvature (depending on the landmark curvature type) is selected as the desired landmark point.

Let $\mathbf{l}_j$ be the coordinates in the image where landmark i is detected and $\mathbf{v}_j$ be the vertex of the mesh associated with the landmark (recall that $\mathbf{v}_j = \mathbf{u}_j \cdot \mathbf{x}^0$). The fitting error of the mesh due to the landmarks is computed as:

$$E_L(\mathbf{x}^0) = \sum_j (\parallel \mathbf{l}_j - \mathbf{u}_j \cdot \mathbf{x}^0 \parallel). \tag{18}$$

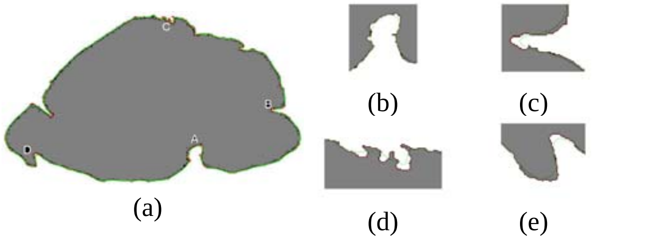


Fig. 18. (a) Traced border of a segmented image before (red) and after (green) smoothing. (b-e) Close-up views of some parts of the boundary marked A-D in (a).

C. Boundary Quads-guided Deformation Energy

For each segment j on the boundary at the kth subdivision level (starting from $k = 1$), the optimal features ${}^{B}F_j^k$ are extracted for the quads Q_j^k on the segment and the classifier parameters ${}^{B}P_j^k$ are used to classify each quad. There are three possibilities when two quads on opposite sides of a crease edge are classified with respect to the atlas. The following rules are used to adjust the boundary in each case:

(i) Classification of both quads is in agreement with the atlas. This happens when the atlas is correctly positioned on the boundary with the region overlaid by the first quad classified as '$+1$' and that overlaid by the second quad classified as '-1' in conformity with the atlas labeling. In this case, there is no need to move the vertices (Fig. 19(a)).

(ii) Either of the quads is classified contrary to the atlas. This typically happens when both quads overlay the same anatomical region and are classified alike, in which case only one will be in agreement with the atlas. The vertices need to be moved in the direction of the quad that is classified contrary to the atlas (Fig. 19(b,c)). In our implementation, the distance to be moved is computed as $0.5|\mathbf{v}'_j - \mathbf{v}_j|$ where $\mathbf{v}'_j$ is the vertex on the wrongly placed quad directly opposite the vertex $\mathbf{v}_j$ on the crease edge between the two regions. This formulation ensures that the maximum distance a vertex can move in a given iteration is the width of the misplaced quad, and this happens only if the adjoining quad is also misclassified.

(iii) Both quads are classified contrary to the atlas. In this case, the two quads are temporarily positioned on either side of the segment and the position that results in correct classification is retained (Fig. 19(d)). If both positions still result in classifications differing from the atlas, the vertices are left unchanged. Note that this condition is quite rare since the mesh boundaries are already fairly close to the image boundaries after global, outer boundary, and landmark-enhanced fitting. The various possibilities are illustrated in Fig. 19(e).

This process is performed iteratively until a specified ratio of the quads (e.g., 95%) are correctly classified, in which case the process is repeated at a finer subdivision level, up to subdivision level 3. With increasing subdivision level, the size of the quads decrease. This results in a progressive reduction in the displacement of the vertices, leading to a smooth fit.

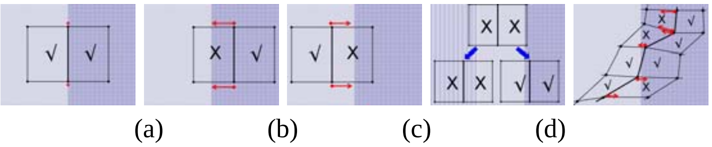


Fig. 19. Classifying opposite quads on a boundary crease edge (the background depicts different regions; check-marks indicate properly classified quads and X indicates misclassified quads): (a) Both are classified in accordance with the model - no displacement of the vertices. (b,c) Displacement of the boundary edge towards the misclassified quad. (d) Rare case of opposite quads being simultaneously incorrectly classified with respect to the model. (e) Various scenarios of quad classification and the resulting displacement at the vertices.

Let the proposed position of each vertex $\mathbf{v}_j$ after an iteration of the boundary fit be $\mathbf{b}_j^k$. The fitting error of the regional boundaries resulting from the classification of the boundary quads at a given subdivision level k is computed as:

$$E_B^k(\mathbf{x}^0) = \sum_j \parallel \mathbf{b}_j^k - \mathbf{u}_j \cdot \mathbf{x}^0 \parallel . \tag{19}$$

D. Mesh Deformation Energy

The deformation energy term $E_d^k(\mathbf{x}^0)$ is the same as that used in Ju *et al.* [20]. It penalizes non-affine deformations of the mesh M^k incurred during the fitting process. Given a mesh consisting of four points p_1, p_2, p_3, p_4 (clockwise order) and the deformed mesh consisting of four corresponding points $\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{p}_4$, the following symmetric penalty term is computed:

$$\frac{a_{234}\hat{p}_1 + a_{134}\hat{p}_2 - a_{124}\hat{p}_3 - a_{123}\hat{p}_4}{(a_{234} + a_{134} + a_{124} + a_{123})^2}, \tag{20}$$

where a_{ijk} denotes the unsigned area of the triangle formed by p_i, p_j, p_k. If we triangulate each quad in the subdivided atlas at level k, taking the sum of the squares of the penalty terms generated by Eq. 20 for all pairs of edge adjacent triangles yields the desired energy term $E_d^k(\mathbf{x}^0)$. To avoid divergence, $E_d^k(\mathbf{x}^0)$ is further normalized by 4^{-k} to account for the growth in the size of the meshes during subdivision.

VI. Results and Discussion

A. Experimental Results

The framework was trained on 36 images manually fitted with subdivision meshes by neuroanatomists and tested on 64 images. For computational efficiency, the images were scaled down to 25% in each dimension before processing. With global affine alignment, the atlas was brought into correspondence with the image, eliminating orientation, translation and scaling differences. Local fitting on the outer boundary further improved the fitting. When the feature-driven landmark detection algorithm was applied, the fitting of internal region boundaries was substantially improved. Figure 20 illustrates typical improvement in the positioning of specified landmarks compared to when only global and outer boundary fitting was used in [20]. To quantify the improvement after incorporating landmark detection, we compared the distance between the 'ground truth' position of each landmark as manually marked

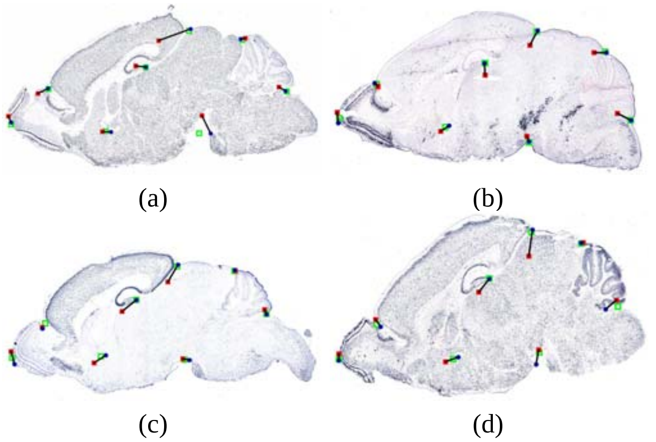


Fig. 20. Typical improvement with the detection of landmark positions in (a) *Sgyr3*, (b) *Th*, (c) *Neurod6*, and (d) *Glur1b* expression images. The red squares are the positions of the landmark vertices after global and outer boundary fitting. The blue circles are the detected landmark positions. The green boxes are expert-annotated ground truth positions.

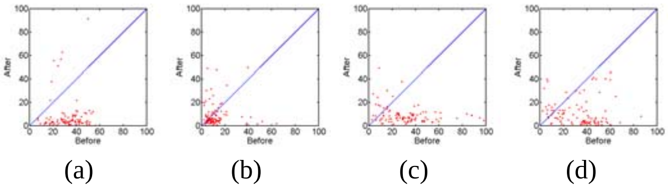


Fig. 21. Differences between the ground truth position of selected landmarks and the position of their vertices before and after incorporating the landmark detection algorithms. Shown are plots for landmarks (a) 2, (b) 5, (c) 7, and (d) 8 in 64 images. The horizontal axis represents the distance (in pixels) between the vertex corresponding to the landmark and the ground truth position after global and outer boundary fitting but before landmark detection. The vertical axis represents the corresponding distance after landmark detection.

by experts and the vertex in the mesh corresponding to the landmark before and after incorporating landmark detection. In Fig. 21, the Euclidean distances of the landmark vertices to the ground truth position are compared. Points below the 45° line represent images where the distance to the ground truth was reduced after landmark detection. It was observed that the distance from the ground truth was reduced in 77%, 94%, 78%, 86%, and 90% of the images for landmarks 0, 2, 5, 7, and 8 respectively. Typical improvements in some specific regions are depicted in Fig. 22(a,b).

To quantify the quality of fit using our hybrid segmentation approach after incorporating the region boundary detector, we computed the spatial overlap between individual anatomical regions as delineated by our framework with those manually delineated by neuroanatomists on the 64 test images. We employed the Dice Similarity Coefficient (DSC) which is widely used in the literature [34], [35], [36] to compare segmentation performance. Using DSC, the spatial overlap for each region was computed as: $r = \frac{2|P_h \cap P_m|}{|P_h|+|P_m|}$ where P_h represents the pixels belonging to the specified region as annotated by the hybrid framework, and P_m is the corresponding set of pixels as annotated by experts. The value of r ranges between 0, indicating no overlap, and 1, indicating complete spatial overlap. In Fig. 23(a), the accuracy of fit (measured by spatial overlap) of all 14 anatomical regions in cross section 9 is

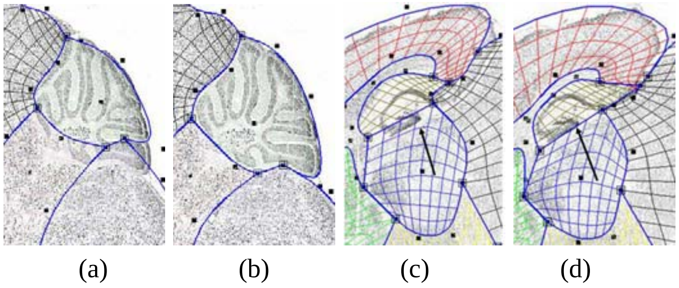


Fig. 22. The cerebellum area of the *Ntrk3* gene expression image overlaid with the mesh fitted (a) before, and (b) after local fitting. The area around the dentate gyrus for the *Calb1* image fitted (c) before, and (d) after landmark/region boundary fitting. The square dots are mesh control points.

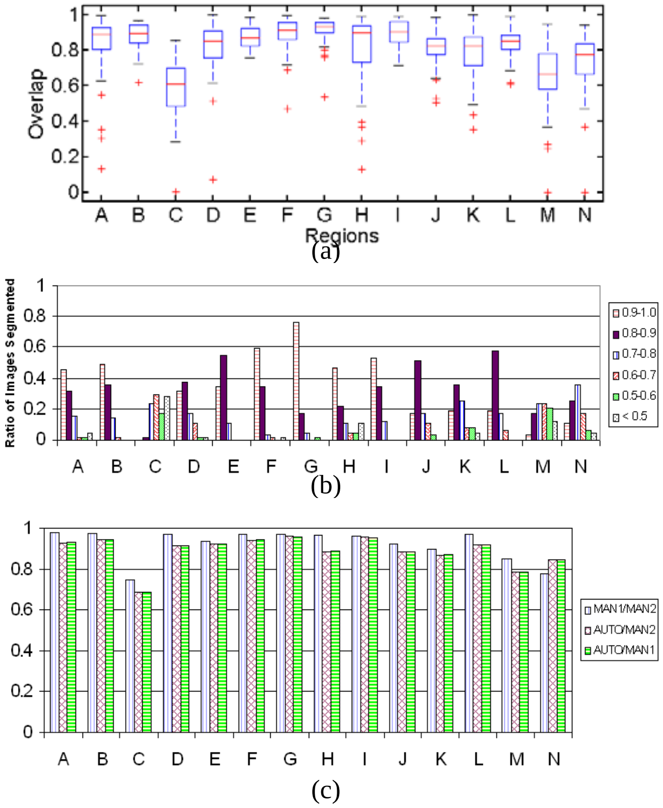


Fig. 23. (a) Boxplot representation of the spatial overlap of manual and automated segmentation for all 14 regions in cross section 9 in 64 expression images. (b) A break down of the fitted images into different ranges of spatial overlap. (c) Comparing the mean spatial overlap between the segmentation by the automated framework (AUTO) and two experts (MAN1, MAN2). The regions are labeled as follows: A-cerebellum, B-cortex, C-basal forebrain, D-hippocampus, E-hypothalamus, F-medulla, G-midbrain, H-olfactory bulb, I-pons, J-septum, K-striatum, L-thalamus, M,N-ventral striatum.

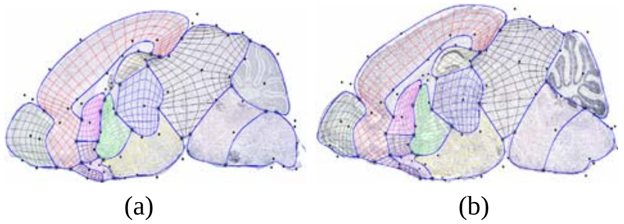


Fig. 24. The result of fitting the standard mesh on (a) *ChAT*, and (b) *BMALI* gene expression images. The dots are mesh control points.

compared for all the images tested. The two sub regions of the ventral striatum are treated as two separate regions for purposes of comparison. The top and bottom of the boxes represent the lower and upper quartiles, while the line inside the box indicates the median overlap for all tested images. The whiskers extending from the boxes show the extent of the remaining data while the asterisks indicate outliers. It can be observed that the median spatial overlap in most regions is above 0.8 (Fig. 23(b)), the highest being the midbrain with a median overlap of about 0.92. A typical example of improvement in region boundary fitting is depicted in Fig. 22(c,d). Examples of fitting using our approach are illustrated in Fig. 24.

To further compare the automated segmentation with manual annotation, experts were requested to re-annotate 30 images belonging to atlas cross section 9. Comparison was then made between the first and second manual annotation (MAN1/MAN2) and also between the automated segmentation and the first and second manual segmentations (AUTO/MAN1 and AUTO/MAN2 respectively). As depicted in Fig. 23(c)) the automated algorithm generally performs well where the manual annotation agrees. There is lower accuracy in regions such as the Basal Forebrain where the manual annotations also have lower spatial overlap.

B. Discussion

The subdivision-based atlas enables powerful user-defined graphical queries, and can ensure that accurate boundaries between major regions are preserved, as demonstrated at `http://geneatlas.org`. This makes subdivision more powerful than a standard warped-template type method [20]. There are two motivating applications for automating this atlas-based segmentation method. The first is to help enable the creation of an accurately annotated searchable database of a large number of gene expression patterns generated using high throughput methods. The second is to facilitate the quantitative comparison of gene expression in experiments, such as between a set of control and experimentally modified specimens [7], [37], [38]. In order to be useful for these applications, the automated method must run in a practical amount of time and segment accurately enough to significantly reduce the amount of human time required.

The segmentation framework detailed here was implemented in Java due to the desire to integrate with an existing online gene expression database application (`http://geneatlas.org`). On a Pentium M 1.66MHz laptop with 1GB RAM, automated segmentation of each tissue section was performed in approximately two minutes. The majority of this time was spent on iterative extraction and classification of features in boundary quads at multiple mesh subdivision levels. For a typical lab with a HITISH data collection system, the throughput is 8 genes per day [4]. For all 11 standard sagittal sections for each of these 8 genes, automated fitting on 88 tissue sections could be performed in less than 3 hours on a modern laptop computer, and thus easily run overnight.

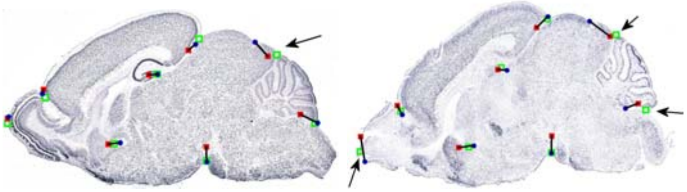


Fig. 25. Examples of wrong landmark detection - (a) landmark 1 in the *Camk2gamma* expression image, and (b) landmarks 1 and 9 in the the *Glur1c* expression image. The red squares are the positions of the landmark vertices after global and outer boundary fitting. The blue circles are the detected landmark positions. The green boxes are expert-annotated ground truth positions.

In regions where the distinction along the boundaries is pronounced such as the cortex, cerebellum and their adjoining regions, we expect to have higher accuracy in delineating the boundaries. Similarly, we expect lower accuracy in regions where this distinction is minimal, such as in the forebrain and the ventral striatum. This is confirmed by our results summarized for all 14 regions in Fig. 23(b). Lower quality automated fitting was encountered in regions with boundaries that are not easily distinguished and lacking landmarks, such as the basal forebrain and the ventral striatum. Examples of failed detection of landmarks are provided in Fig. 25. In Fig. 25(a), there is an artifact in the expected position of landmark 1 (marked by an arrow) thus affecting the value of the curvature. A nearby location with concave curvature was detected instead. A similar situation is depicted in Fig. 25(b) for landmarks 1 and 9. This suggests that the performance could be improved with the identification of additional reliable landmarks. In some cases where the background signal was very weak, neither the landmarks nor the region boundaries were accurately detected. In these cases, deformations relied only on the fitting of external boundaries. Performance could thus be improved by developing a data production protocol with a more consistently visible background signal. The results show that the automated segmentation method described here is successful in achieving a reasonably accurate fit of the atlas in the majority of cases. This is a substantial improvement over a method relying on boundary fitting alone. A more accurate automated fitting should reduce the time required for any manual intervention afterwards. Manual fitting should always be applied to verify and correct. Due to the wide variation in the data, manual intervention will likely always be part of the process. However, reducing the amount of human time required for large scale analysis is a highly desirable outcome.

The quality of fit is generally suitable for the desired applications. In most studies and database queries, a region of interest is defined. Accuracy thus depends on the atlas mapping to different specimens overlapping in that region of interest, and the boundaries being reasonably defined. Two other data collection issues limit the performance of the method. The first is damage to the tissue during collection and subsequent slicing. The second is the angle of slicing, which when off by a few degrees can cause the tissue section to not properly match the regional topology in the 2D atlas. There are thus limitations, even in manually fit data, to how accurately this whole-slice subdivision mesh can represent small anatomical sub-regions. In these cases, one would want to create a specialized subdivision mesh. For example, for research primarily interested in sub-thalamic nuclei, one would create a new subdivision mesh atlas focused on the thalamus, with the boundaries of the nuclei explicitly included as part of the mesh. The atlas would be trained to fit this thalamic dataset.

The next step in the further development of this method is to extend the technique into 3D, where automated fitting will become a virtual necessity. By accurately stacking the tissue sections collected [39], and creating a 3D atlas of the brain [10], all tissue sections from the brain could be included in the database for experimental comparison, rather than a standard subset of slices. In addition, the issue of variations in sectioning angle between brains would be eliminated, thus potentially increasing the accuracy of automated fitting. The challenge will be optimizing the fitting technique to handle the additional amount of data being included in a reasonable time. In addition, in the cases where different genes are marked in alternating slices from the same brain, a signal normalization will be required for accurate stacking of the tissue sections. A 3D version of this method could be applied to micro MRI data of the brain, or be used to help bring mixed-model data into the same spatial context. The method is not limited to the brain. It could conceivably be applied to many different anatomical and non-anatomical applications of a subdivision atlas where multiresolution spatial databases and comparisons are desired.

VII. Conclusion

We have presented a new method for the automated segmentation of tissue images revealing gene expression patterns using a subdivision-based anatomical atlas of the mouse brain. Our learning framework utilizes statistical shape information, appearance of anatomical landmarks, and texture variation at anatomical boundaries to fit the atlas to given expression images. We applied the framework to a set of sagittal tissue sections of 100 different gene expression patterns from the P7 mouse brain. The framework is automatic, fast, and accurate - three major requirements for justifying its integration into applications such as creating a database of gene expression patterns and performing comparative analysis of patterns. By automating the segmentation of gene expression, this method can play a key role in helping scientists answer questions about gene functions and disease mechanisms.

References

[1] R. Waterston and the Mouse Genome Sequencing Consortium, "Initial sequencing and comparative analysis of the mouse genome," *Nature*, vol. 420, pp. 520–562, December 2002.

[2] U. Herzig, C. Cadenas, F. Sieckmann, W. Sierralta, C. Thaller, A. Visel, and G. Eichele, "Development of high-throughput tools to unravel the complexity of gene expression patterns in the mammalian brain," in *Complexity in Biological Information Processing*, ser. Novartis Foundation Symposium 239, G. Bock and J. Goode, Eds. Chicester: John Wiley & Sons, 2001, pp. 129–149.

[3] U. Albrecht, H.-C. Lu, J.-P. Revelli, X.-C. Xu, R. Lotan, and G. Eichele, *Studying Gene Expression on Tissue Sections Using In Situ Hybridization*. Boca Raton: CRC Press, 1997, pp. 93 – 119.

[4] J. Carson, C. Thaller, and G. Eichele, "A transcriptome atlas of the mouse brain at cellular resolution," *Curr Opin Neurobiol*, vol. 12, no. 5, pp. 562–565, Oct 2002.

[5] A. Visel, C. Thaller, and G. Eichele, "Genepaint.org: an atlas of gene expression paterns in the mouse embryo," *Nucleic Acids Research*, vol. 32, pp. D552–556, 2004.

[6] L. Ng, M. Hawrylycz, and D. Haynor, "Automated high-throughput registration for localizing 3D mouse brain gene expression using ITK," in *Proc. MICCAI Workshop on Open-Source Software*, Palm Springs, CA, October 2005.

[7] J. Carson, "Quantitative annotation and analysis of gene expression patterns with an atlas of the mouse brain," Ph.D. dissertation, Baylor College of Medicine, January 2004.

[8] W. Ali and F. Cohen, "Registering coronal histological sections of a rat brain with coronal sections of a 3D brain atlas using geometric curve invariants and b-spline representation," *IEEE Transactions on Medical Imaging*, vol. 17, no. 6, pp. 957–966, December 1998.

[9] S. Wirtza, B. Fischera, J. Modersitzkib, and O. Schmitt, "Super-fast elastic registration of histologic images of a whole rat brain for three-dimensional reconstruction," in *Medical Imaging 2004: Image Processing. Proceedings of the SPIE*, J. Fitzpatrick and M. Sonka, Eds., vol. 5370, 1998, pp. 328–334.

[10] T. Ju, J. Warren, J. Carson, G. Eichele, C. Thaller, W. Chiu, M. Bello, and I. Kakadiaris, "Building 3D surface networks from 2D curve networks with application to anatomical modeling," *The Visual Computer*, vol. 21, no. 8, pp. 764–773, September 2005. [Online]. Available: http://www.springerlink.com/content/q1241849480113q8

[11] The High Resolution Mouse Brain Atlas at Harvard, "http://www.hms.harvard.edu/research/brain."

[12] C. Glasbey and K. Mardia, "A review of image warping methods," *Journal of Applied Statistics*, vol. 25, pp. 155–171, 1998.

[13] Mouse Atlas Project, "http://www.loni.ucla.edu/map/index.html."

[14] J. Warren and H. Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach*. San Francisco, CA: Morgan Kaufmann Publishers, 2002.

[15] C. Mandal, B. C. Vemuri, and H. Qin, "A new dynamic FEM-based subdivision surface model for shape recovery and tracking in medical images," in *Proc. First International Conference on Medical Image Computing and Computer-Assisted Intervention*. London, UK: Springer-Verlag, 1998, pp. 753–760.

[16] C. Mandal, H. Qin, and B. C. Vemuri, "Dynamic modeling of butterfly subdivision surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 3, pp. 265–287, 2000.

[17] S. Pizer, T. Fletcher, Y. Fridmana, D. Fritsch, A. Gash, J. Glotzer, S. Joshi, A. Thall, G. Tracton, P. Yushkevich, and E. Chaney, "Deformable m-reps for 3D medical image segmentation," *International Journal of Computer Vision*, vol. 55, no. 2, pp. 85 – 106, November 2003.

[18] I. Kakadiaris, M. Bello, S. Arunachalam, W. Kang, T. Ju, J. Warren, J. Carson, W. Chiu, C. Thaller, and G. Eichele, "Landmark-driven, atlas-based segmentation of mouse brain tissue images containing gene expression data," in *Proc. 7th International Conference on Medical Image Computing and Computer-Assisted Intervention*, Rennes, France, September 2004, pp. 192–199.

[19] M. Bello, T. Ju, J. Warren, J. Carson, W. Chiu, C. Thaller, G. Eichele, and I. Kakadiaris, "Hybrid segmentation framework for tissue images containing gene expression data," in *Proc. 8th International Conference on Medical Image Computing and Computer-Assisted Intervention*, Palm Springs, CA, October 2005, pp. 254–261.

[20] T. Ju, J. Warren, G. Eichele, C. Thaller, W. Chiu, and J. Carson, "A geometric database for gene expression data," in *Proc. Eurographics Symposium on Geometry Processing*, L. Kobbelt, P. Schröder, and H. Hoppe, Eds., Aachen, Germany, June 2003, pp. 166 – 176.

[21] F. Valverde, *Golgi atlas of the postnatal mouse brain*. New York: Springer-Verlag, 1998.

[22] E. Catmull and J. Clark, "Recursively generated b-spline surfaces on arbitrary topological meshes," *Computer-Aided Design*, vol. 16, no. 6, pp. 350–355, 1978.

[23] B. van Ginneken, A. Frangi, J. Staal, B. ter Haar Romeny, and M. Viergever, "Active shape model segmentation with optimal features," *IEEE Transcations on Medical Imaging*, vol. 21, no. 8, August 2002.

[24] S. Aksoy and R. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 563 – 582, May 2001.

[25] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157 – 1182, March 2003.

[26] P. Radivojac, Z. Obradovic, K. Dunker, and S. Vucetic, “Feature selection filters based on the permutation test,” in *Proc. 15th European Conference on Machine Learning*, Pisa, Italy, September 2004, pp. 334 – 346.

[27] U. Fayyad and K. Irani, “On the handling of continuous-valued attributes in decision tree generation,” *Machine Learning*, vol. 8, no. 1, pp. 87 – 102, January 1992.

[28] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. John Wiley & Sons Inc., 2001.

[29] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 2000.

[30] C. Hsu, C. Chang, and C. Lin, “A practical guide to support vector classification,” National Taiwan University, Tech. Rep., July 2003.

[31] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[32] K. Laws, “Textured image segmentation,” Ph.D. dissertation, USC, 1980.

[33] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, 2nd ed. Pacific Grove, USA: Brooks/Cole Publishing Company, 1999.

[34] K. H. Zou, S. K. Warfield, A. Bharatha, C. M. Tempany, M. R. Kaus, S. J. Haker, W. M. Wells, F. A. Jolesz, and R. Kikinis, “Statistical validation of image segmentation quality based on a spatial overlap index,” *Academic Radiology*, vol. 11, no. 2, pp. 178–189, February 2004.

[35] S. M. Pizer, P. T. Fletcher, S. Joshi, A. Thall, J. Z. Chen, Y. Fridman, D. S. Fritsch, A. G. Gash, J. M. Glotzer, M. R. Jiroutek, C. Lu, K. E. Muller, G. Tracton, P. Yushkevich, and E. L. Chaney, “Deformable m-reps for 3d medical image segmentation,” *Int. J. Comput. Vision*, vol. 55, no. 2-3, pp. 85–106, 2003.

[36] V. Grau, A. U. J. Mewes, M. Alcaiz, R. Kikinis, and S. K. Warfield, “Improved watershed transform for medical image segmentation using prior information,” *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 447–458, April 2004.

[37] J. Carson, T. Ju, H.-C. Lu, C. Thaller, M. Xu, S. L. Pallas, M. A. Crair, J. Warren, W. Chiu, and G. Eichele, “A digital atlas to characterize the mouse brain transcriptome,” *PLoS Computational Biology*, vol. 1, no. 4, p. e41, 2005.

[38] B. McGill, S. Bundle, M. Yaylaoglu, J. Carson, C. Thaller, and H. Zoghbi, “Enhanced anxiety and stress-induced corticosterone release are associated with increased crh expression in a mouse model of rett syndrome,” *PNAS*, vol. 103, no. 48, pp. 18 267–18 272, 2006.

[39] T. Ju, J. Warren, J. Carson, M. Bello, I. Kakadiaris, C. Thaller, and G. Eichele, “3D volume reconstruction of a mouse brain from histological sections using warp filtering,” *Journal of Neuroscience Methods*, vol. 156, no. 1-2, pp. 84–100, September 2006.