Curve Sampling and Geometric Conditional Simulation

Ayres Fan

Joint work with John Fisher, William Wells,
Jonathan Kane, and Alan Willsky

Stochastic Systems Group
Massachusetts Institute of Technology

September 19, 2007

Outline of the talk

1. Curve evolution and MCMC methods.
2. MCMC curve sampling.
3. Visualizing curve samples.
4. Conditional simulation.
5. Hybrid 2D/3D models.
6. Conclusions and future work

Curve Evolution

- Given an image I defined on an image domain $\Omega \subset \mathbb{R}^2$, curve evolution methods attempt to find a curve $\vec{C} : [0, 1] \to \Omega$ that best segments it.

- This is typically done by constructing an energy functional $\mathrm{E}(\vec{C})$ (which can be viewed as a negative log probability distribution) and finding a local minimum using gradient descent.

- This results in a geometric gradient flow which can expressed in terms of a force f times the normal to the curve $\vec{\mathcal{N}}_{\vec{C}}$:

$$\frac{\mathrm{d}\vec{C}}{\mathrm{d}t}(p) = f(p)\vec{\mathcal{N}}_{\vec{C}}(p) \quad .$$

Level Set Methods

- A natural numerical implementation to track $\vec{C}$ is to use marker points on the boundary. This approach has problems with reinitialization and topological change.
- Level sets are an alternative approach which evolve a surface Ψ (one dimension higher than our curve) whose zeroth level set is $\vec{C}$. (Osher and Sethian 1988)
- Standard curve derivatives can be written in terms of Ψ:

$$\vec{\mathcal{N}}_{\vec{C}} = \frac{\nabla \Psi}{\|\nabla \Psi\|} \text{ and } \kappa_{\vec{C}} = \nabla \cdot \left(\frac{\nabla \Psi}{\|\nabla \Psi\|} \right) \quad .$$

- Setting $\Psi(\vec{C}(p)) = 0$ for all $p \in [0,1]$ and differentiating with respect to t, we obtain:

$$\frac{\mathrm{d}\Psi}{\mathrm{d}t} = \frac{\mathrm{d}\vec{C}}{\mathrm{d}t} \cdot \nabla \Psi = f(\vec{\mathcal{N}}_{\vec{C}} \cdot \nabla \Psi) = f\|\nabla \Psi\| \quad .$$

Markov Chain Monte Carlo

- Markov Chain Monte Carlo (MCMC) methods are a class of algorithms which are designed to generate samples from a target distribution $\pi(x)$.
- $\pi(x)$ is difficult to sample from directly, so instead a Markov chain with transition probability $\mathrm{T}(y\,|\,x)$ is constructed whose stationary distribution is $\pi(x)$:

$$\pi(z) = \int \pi(x)\mathrm{T}(z\,|\,x)\mathrm{d}x \ .$$

- Detailed balance is a sufficient condition for this to hold:

$$\pi(z)\mathrm{T}(x\,|\,z) = \pi(x)\mathrm{T}(z\,|\,x) \ .$$

- If a chain is ergodic and detailed balance holds, successive samples from $\mathrm{T}(z\,|\,x)$ asymptotically become samples from $\pi(x)$.

Metropolis-Hastings

- General method developed by Metropolis *et al.* (1953) and extended by Hastings (1970).
- Define transition probability as the product of a proposal distribution $\mathtt{q}(y \,|\, x)$ and an acceptance probability $\mathtt{a}(y \,|\, x)$.
- A candidate sample is generated from $\mathtt{q}$, and the Hastings ratio is computed:

$$\eta(y \,|\, x) = \frac{\pi(y)\mathtt{q}(x \,|\, y)}{\pi(x)\mathtt{q}(y \,|\, x)} \quad .$$

- Then $z = y$ with probability $\min(1, \eta(y \,|\, x))$. Otherwise $z = x$.
- Problem of sampling from π is now the problem of generating many samples from $\mathtt{q}$ and evaluating π.

Gibbs Sampling

- MCMC method developed by Geman and Geman (1984). Most easily applied to models which have a Markov structure.
- Begin by dividing the variables into two subsets x_S and $x_{\setminus S}$. $x_{\setminus S}$ remains unchanged (so $y_{\setminus S} = x_{\setminus S}$).
- The proposal distribution $\mathtt{q}(y_S \,|\, x)$ is defined to be the conditional probability of y_S given the remaining variables: $\mathtt{q}(y_S \,|\, x) = \pi(y_S \,|\, x_{\setminus S})$. The resulting sample is always accepted, so $\mathtt{a}(y \,|\, x) = 1$.
- If the model is defined by a Markov graph structure, $\pi(y_S \,|\, x_{\setminus S}) = \pi(y_S \,|\, x_{\mathcal{N}(S)})$ where $\mathcal{N}(S)$ is the neighborhood of S.
- The subset S changes over time. This can be done randomly or according to some deterministic sequence.

Curve Sampling

- We construct a curve sampling framework which generates samples based on random curve perturbations.
- There are a number of benefits of sampling over traditional optimization-based curve evolution:
  - Naturally handles multi-modal distributions
  - Can help avoid local minima
  - Higher-order statistics (*e.g.*, error variances)
  - Conditional simulation

Curve Perturbations

- For consistency, all perturbations for a curve $\vec{C}$ are defined relative to a canonical arc length curve parameterization $\vec{C}_{\mathrm{a}}$.
- Generate random, correlated Gaussian noise

$$f^{(t+1)} = \mu_{\vec{C}_{\mathrm{a}}^{(t)}}(p) + h \circledast n^{(t+1)}(p) \quad .$$

- These smooth perturbation are added to the normal of the curve:

$$\vec{\Gamma}^{(t+1)}(p) = \vec{C}_{\mathrm{a}}^{(t)}(p) + f^{(t+1)}(p)\vec{\mathcal{N}}_{\vec{C}_{\mathrm{a}}^{(t)}}(p)\delta t$$

- Our standard choice for μ is:

$$\mu_{\vec{C}_{\mathrm{a}}^{(t)}}(p) = -\alpha\kappa_{\vec{C}_{\mathrm{a}}^{(t)}}(p) + \gamma_{\vec{C}_{\mathrm{a}}^{(t)}} \quad .$$

Detailed Balance

- To implement Metropolis-Hastings, we need to be able to calculate the Hastings ratio:

$$\eta(\vec{\Gamma}^{(t+1)} \mid \vec{C}^{(t)}) = \frac{\pi(\vec{\Gamma}^{(t+1)})\mathsf{q}(\vec{C}^{(t)} \mid \vec{\Gamma}^{(t+1)})}{\pi(\vec{C}^{(t)})\mathsf{q}(\vec{\Gamma}^{(t+1)} \mid \vec{C}^{(t)})} \quad .$$

- The probability of the forward perturbation is:

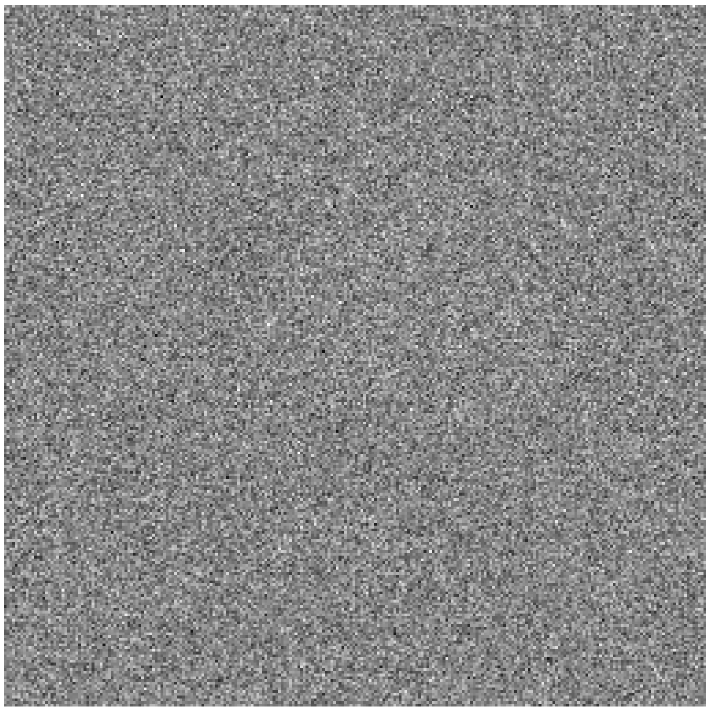
$$\mathsf{q}(\vec{\Gamma}^{(t+1)} \mid \vec{C}^{(t)}) = \mathrm{p}(\boldsymbol{f}) \propto \exp\left(-\frac{\boldsymbol{n}^{\mathrm{T}}\boldsymbol{n}}{2\sigma^2}\right)$$

- Reverse perturbation is:

$$\vec{C}_{\mathrm{r}}(q) = \vec{\Gamma}_{\mathrm{a}}(q) + \phi(q)\vec{\mathcal{N}}_{\vec{\Gamma}_{\mathrm{a}}}(q)\delta t$$

$$\phi(q) = \mu_{\vec{\Gamma}_{\mathrm{a}}}(q) + h \circledast \nu(q)$$

Synthetic Noisy Image Example



- Assume a piecewise-constant image $m(\boldsymbol{x})$ with white Gaussian noise $w(\boldsymbol{x})$:

$$I(\boldsymbol{x}) = m(\boldsymbol{x}) + w(\boldsymbol{x})$$

- This corresponds to the Chan-Vese energy functional (which also adds a regularizing term):

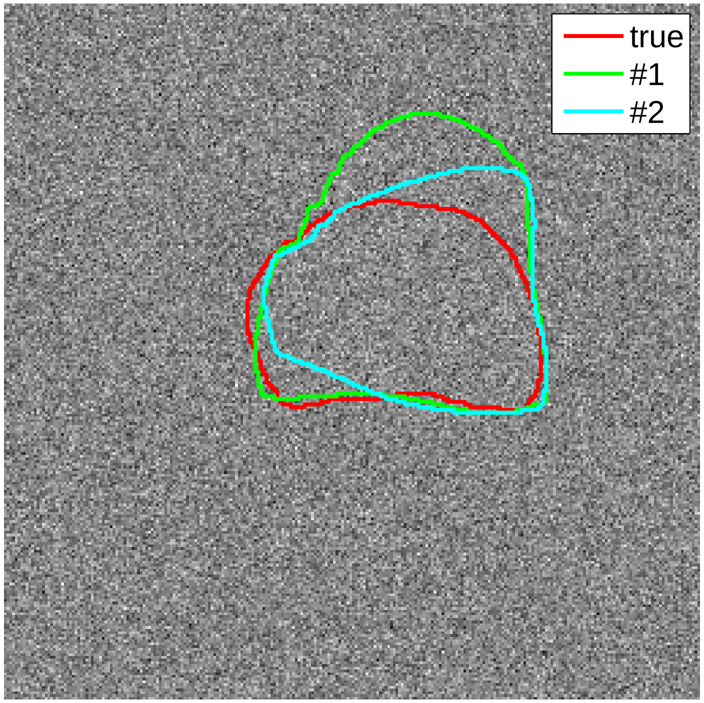
$$\mathrm{E}(\vec{C}) = \iint\limits_{\mathcal{R}_{\vec{C}}} (I - m_1)^2 \mathrm{d}\boldsymbol{x} + \iint\limits_{\mathcal{R}^c_{\vec{C}}} (I - m_0)^2 \mathrm{d}\boldsymbol{x} + \beta \oint\limits_{\vec{C}} \mathrm{d}s$$

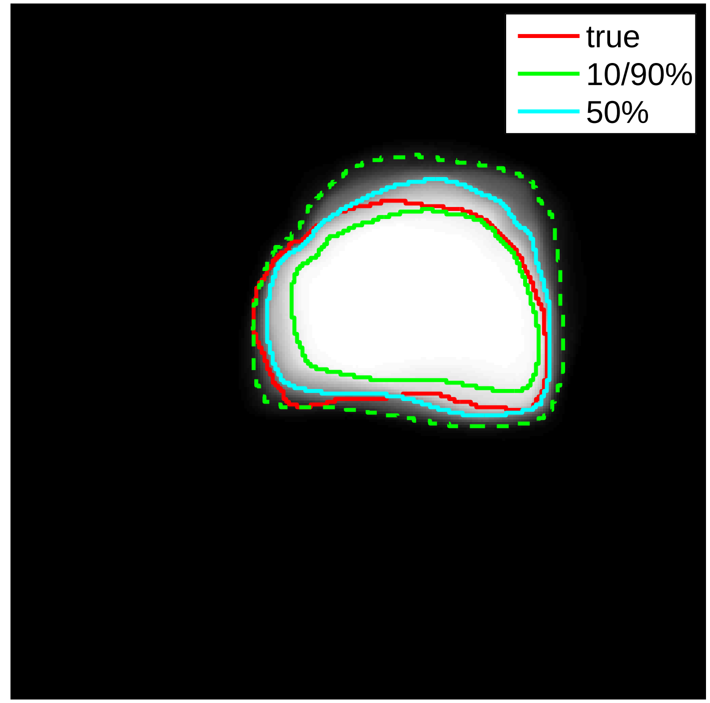
Visualizing Samples

We use three main methods for visualizing the output of our curve sampling algorithm:

1. Most likely samples
   - Close to the global maximum.
2. Histogram images
   - Given samples $\{\vec{C}_n\}_{n=1}^{N}$, $\Phi(\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{H}(-\Psi_{\vec{C}_N}(\boldsymbol{x}))$ (*i.e.*, the percentage of samples for which $\boldsymbol{x}$ is inside the curve).
3. Marginal confidence bounds
   - Level curves of Φ. Contours which can give an idea of the range of likely locations of the true curve.

Synthetic Gaussian Results





- Most likely samples not very accurate (due to specific noise configuration).
- 10/90% confidence bounds bracket the true answer.

Conditional Simulation

- In many problems, the model admits many reasonable solutions. This can be due to a low signal-to-noise ratio (SNR) or an ill-posed estimation problem.
- For most segmentation algorithms, user input is limited to initialization and parameter selection.
- *Conditional simulation* involves sampling part of the solution conditioned on the rest being known (*e.g.*, pinned Brownian motion).
  - For curve sampling, part of the curve is specified. Much more feasible for sampling than constrained optimization in high-dimensional spaces.
  - Can help with both accuracy and convergence speed.
  - Leads to interactive semi-automatic segmentation approaches.

Conditional Curve Sampling

- Let $\vec{C}_{\mathrm{k}} : [0, b] \to \Omega$ be the known portion of the curve, and $\vec{C}_{\mathrm{u}} : [b, 1] \to \Omega$ be the unknown portion.
- We now wish to sample from $\tilde{\pi}(\vec{C}_{\mathrm{u}} \,|\, \vec{C}_{\mathrm{k}})$:

$$\tilde{\pi}(\vec{C}_{\mathrm{u}} \,|\, I, \vec{C}_{\mathrm{k}}) \propto \mathrm{p}(I \,|\, \vec{C}_{\mathrm{u}}, \vec{C}_{\mathrm{k}})\mathrm{p}(\vec{C}_{\mathrm{u}} \,|\, \vec{C}_{\mathrm{k}}) = \mathrm{p}(I \,|\, \vec{C})\mathrm{p}(\vec{C}_{\mathrm{u}} \,|\, \vec{C}_{\mathrm{k}})$$

- We note that $\mathrm{p}(\vec{C}_{\mathrm{u}} \,|\, \vec{C}_{\mathrm{k}}) = \mathrm{p}(\vec{C}_{\mathrm{u}}, \vec{C}_{\mathrm{k}})/\mathrm{p}(\vec{C}_{\mathrm{k}})$, and the denominator can generally only be obtained from $\mathrm{p}(\vec{C})$ by integrating out $\vec{C}_{\mathrm{u}}$.

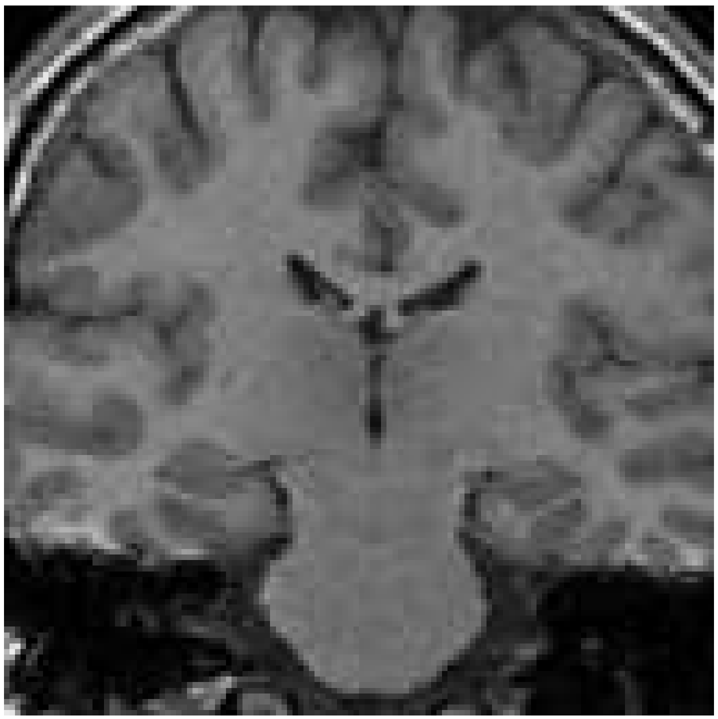
Exact Curve Information

- For special cases, evaluation of $\mathrm{p}(\vec{C}_{\mathrm{u}} \,|\, \vec{C}_{\mathrm{k}})$ is tractable:
  - $\vec{C}$ is low-dimensional.
  - $\vec{C}_{\mathrm{k}}$ is assumed to be exact.
  - $\mathrm{p}(\vec{C})$ has special form (*e.g.*, Markov structure).
- When the curve is specified exactly, we observe that

$$\tilde{\pi}(\vec{C}_{\mathrm{u}} \,|\, I, \vec{C}_{\mathrm{k}}) \propto \mathrm{p}(I \,|\, \vec{C})\mathrm{p}(\vec{C}_{\mathrm{u}}, \vec{C}_{\mathrm{k}})/\mathrm{p}(\vec{C}_{\mathrm{k}}) \propto \pi(\vec{C} \,|\, I)$$

- Thus we see that evaluation of the target distribution is unchanged (except some of the curve does not change with time). The proposal distribution must be modified so that perturbed curves remain on the manifold of curves which contain $\vec{C}_{\mathrm{k}}$.
- To do so, we can multiply our earlier perturbation $f(p)$ by a scalar field $d(p)$ which is 0 on $[0, b]$.

Thalamus Segmentation



- The thalamus is a subcortical brain structure.
- Low-contrast makes it difficult to distinguish it from surrounding cerebral tissue.
- One approach to make the problem better-posed is using shape models (Pohl *et al.* 2004).
- We apply our conditional simulation approach which requires much less training and allows more user control over the segmentation process.

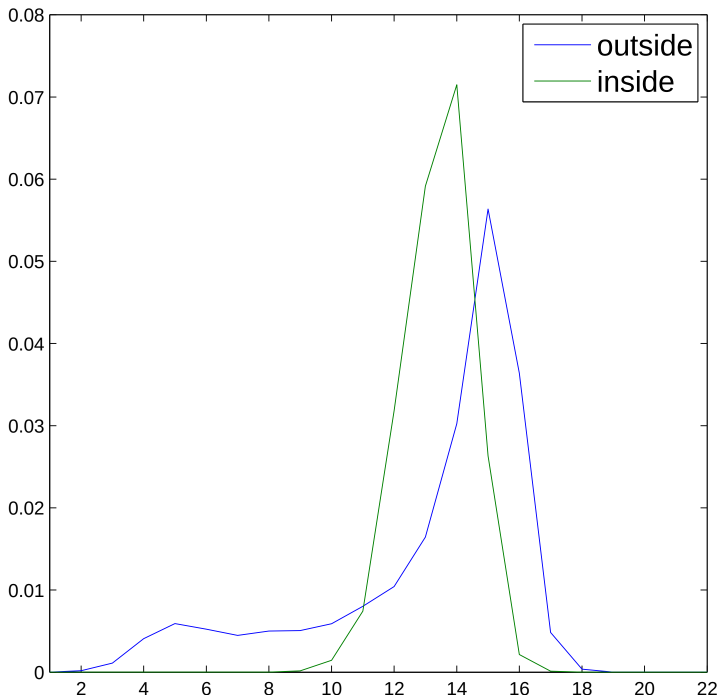
Multiple Disjoint Regions

- Disjoint regions leads to a set of multiple curves $\boldsymbol{C} = \{\vec{C}_i\}_{i=1}^{N_c}$.
- Perturb each curve individually with

$$\mathsf{q}(\boldsymbol{\Gamma} \,|\, \boldsymbol{C}) = \prod_i \mathsf{q}_i(\vec{\Gamma}_i \,|\, \vec{C}_i) \quad .$$

- Curves are coupled together through the evaluation of π. Because pixel intensities in both halves of the thalamus are drawn from the same distribution, we combine the curves into a joint label map $\lambda_{\boldsymbol{C}}(\boldsymbol{x})$ which is 1 if $\boldsymbol{x}$ is inside any $\vec{C}_i$.
- If curves represent objects with different statistics, need to resolve ambiguities caused by overlap.

Thalamus Model



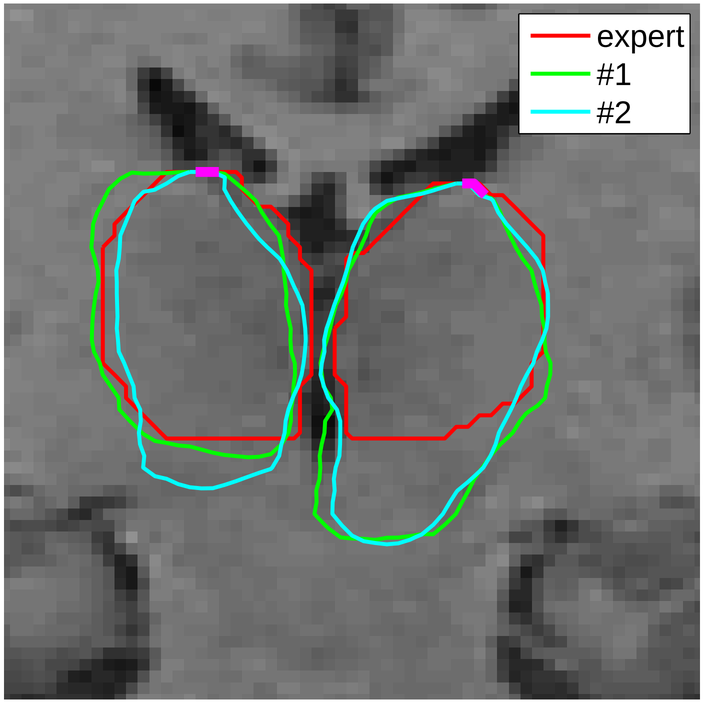
- Learn histograms from band of pixels within a distance d_0 of the expert-segmented boundary.
- Resulting data likelihood:

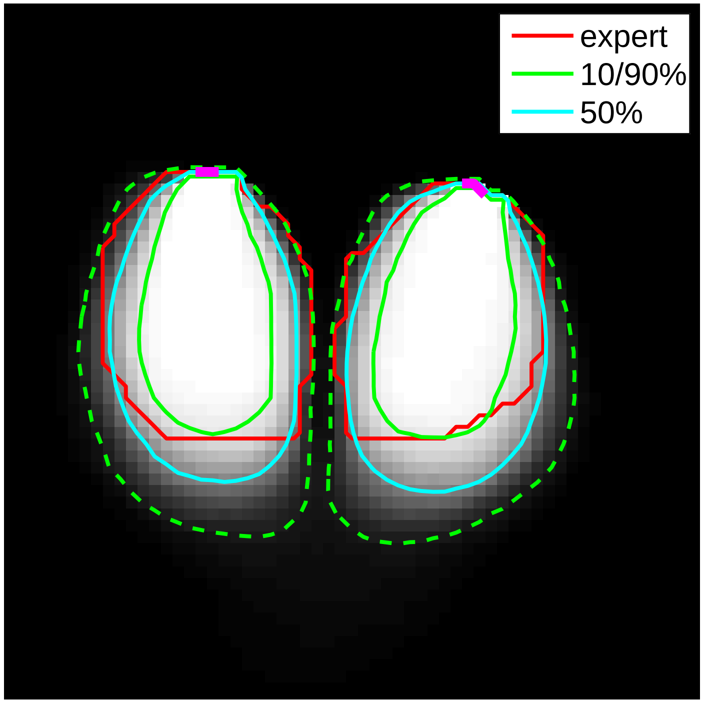
$$\mathrm{p}(I \,|\, \boldsymbol{C}) = \prod_{\substack{\{\boldsymbol{x} \,|\, \exists\, i \text{ s.t.} \\ |\tilde{\Psi}_{\vec{C}_i}(\boldsymbol{x})| \le d_0\}}} \mathrm{p}\left(I(\boldsymbol{x}) \,|\, \lambda_{\boldsymbol{C}}(\boldsymbol{x})\right) \quad .$$

- This leads to an overall target distribution of:

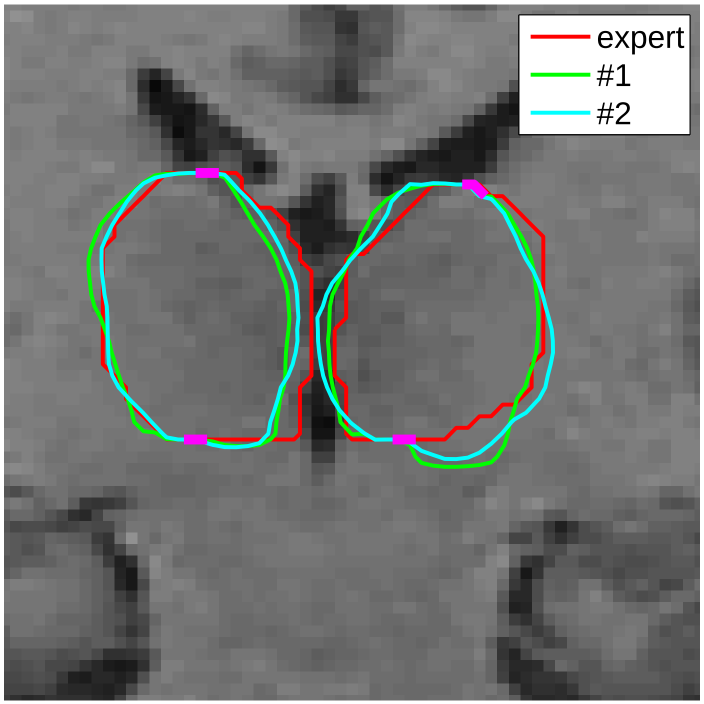
$$\pi(\boldsymbol{C}) \propto \mathrm{p}(I \,|\, \boldsymbol{C}) \exp\left(-\alpha \sum_i \oint_{\vec{C}_i} \mathrm{d}s\right) \quad .$$

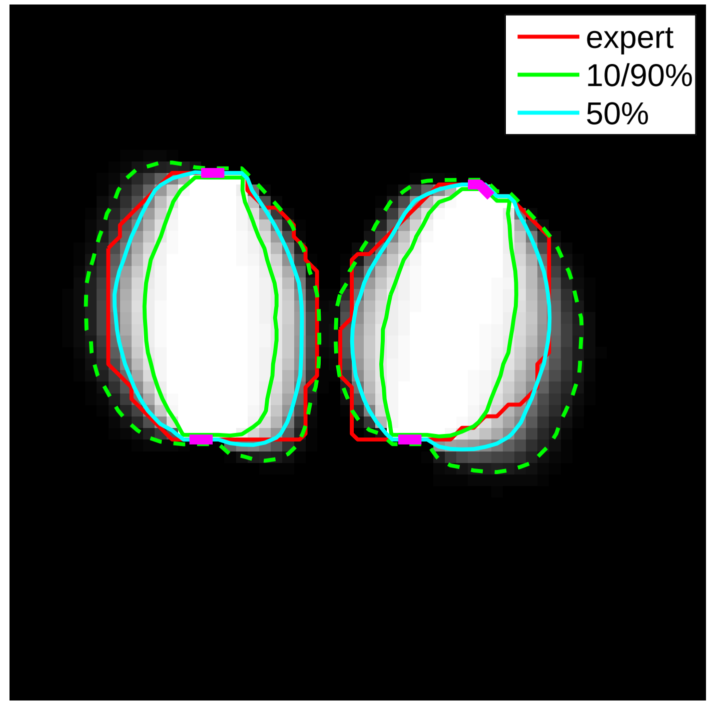
Top Points Fixed



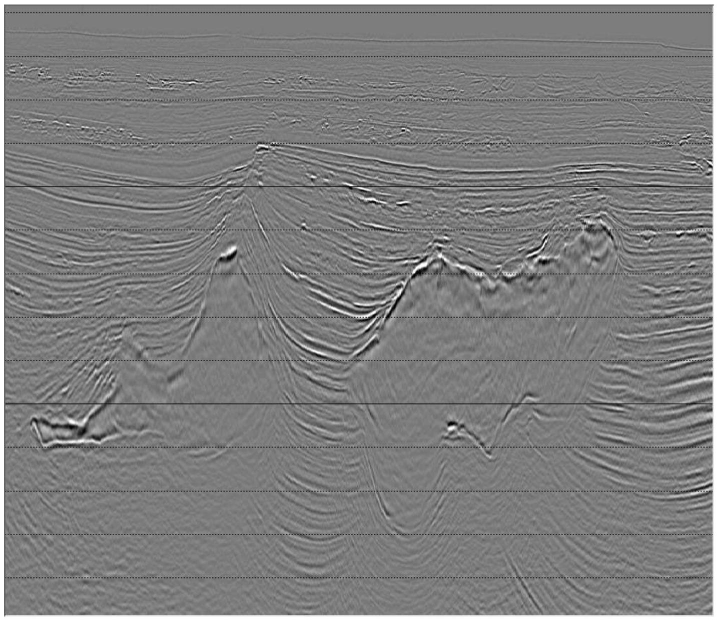


Adding Constraints at the Bottom





Gravity Inversion



- The goal is to find salt body boundaries using an array of surface gravimeters.
- Difficult to image below salt without knowing bottom salt location. Salt bodies also act as liquid traps (*e.g.*, gas, oil).
- Data are processed to remove base effects (*e.g.*, the geoid, centrifugal force) to leave residual gravity effects from differing salt density:

$$\vec{g}_i = G \int_{\Omega} \frac{\rho(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{x}_i)}{\|\boldsymbol{x} - \boldsymbol{x}_i\|^3} \mathrm{d}\boldsymbol{x}$$

Gravity Inversion Model

- To construct a curve-based gravity model, we assume constant density inside and outside salt:

$$\rho(\boldsymbol{x};\vec{C}) = \Delta\rho\mathcal{H}(-\Psi_{\vec{C}}(\boldsymbol{x})) \ .$$

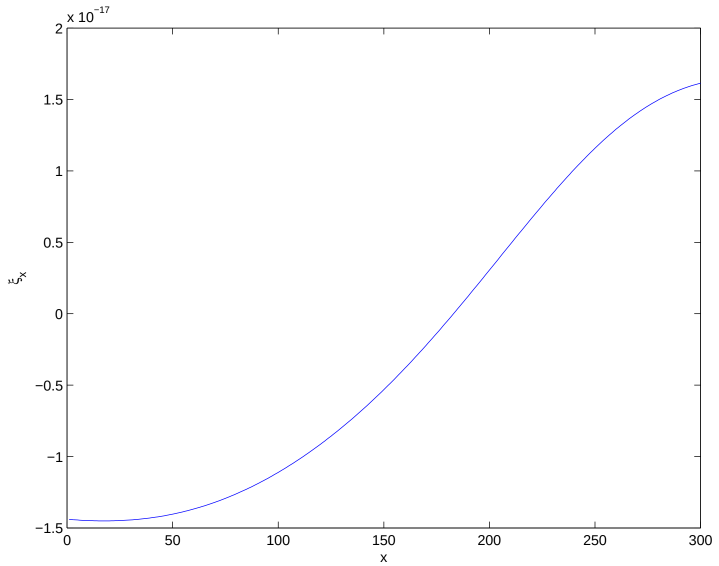
- This leads to the following forward model to translate a curve into a gravity measurement:

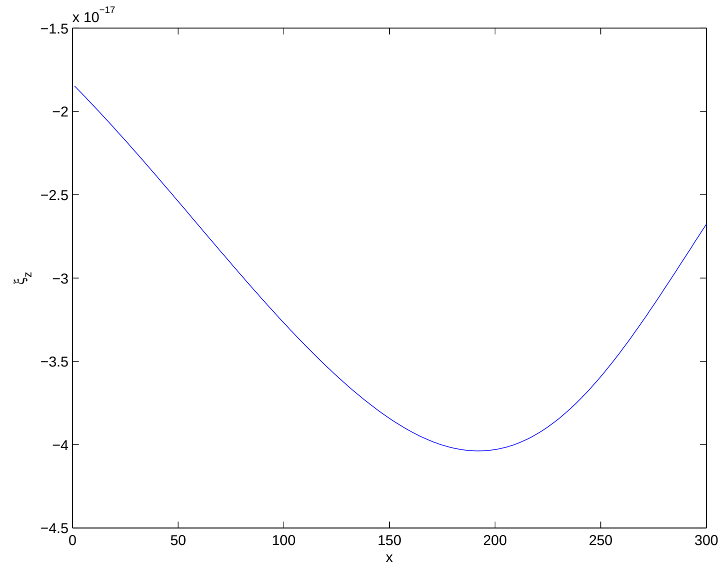
$$\vec{g}_i(\vec{C}) = \Delta\rho G \int_{\mathcal{R}_{\vec{C}}} \frac{(\boldsymbol{x}-\boldsymbol{x}_i)}{\|\boldsymbol{x}-\boldsymbol{x}_i\|^3}\mathrm{d}\boldsymbol{x} \ .$$

- We construct an energy functional that penalizes the L2 error between the observed gravity and the forward model plus a regularization penalty:

$$E(\vec{C}) = \sum_{i=1}^{N_g} ||\vec{g}_i(\vec{C}) - \vec{\xi}_i||^2 + \alpha \oint_{\vec{C}} \mathrm{d}s$$

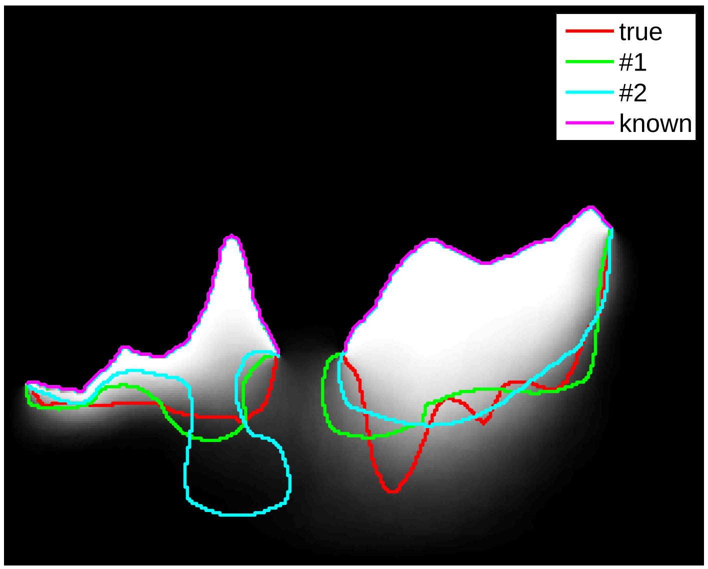
Real Geometry: Gravity Profile

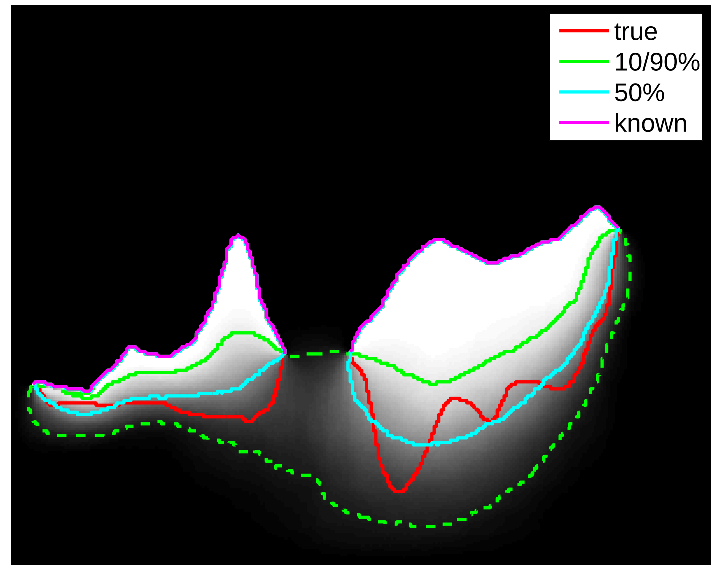




- 600 measurements, 300×240 image (72,000 pixels).
- Synthetic salt body constructed from expert-segmented seismic image.

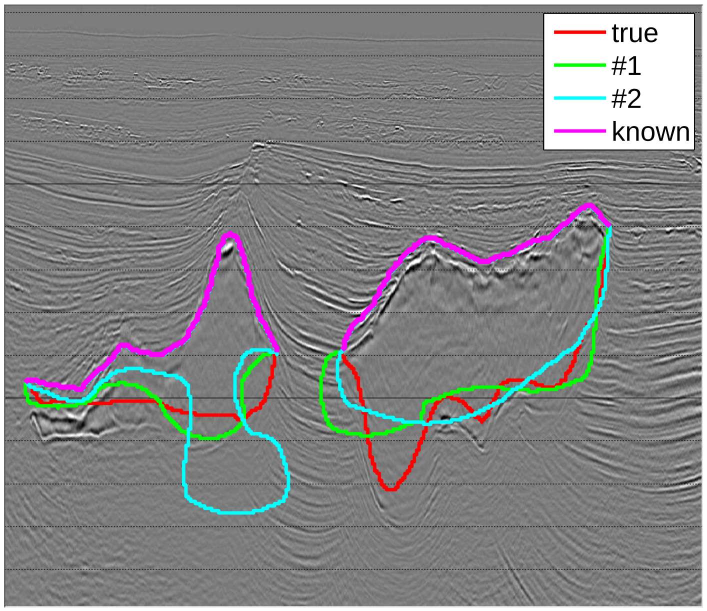
Real Geometry: Most Probable Samples & Confidence Bounds

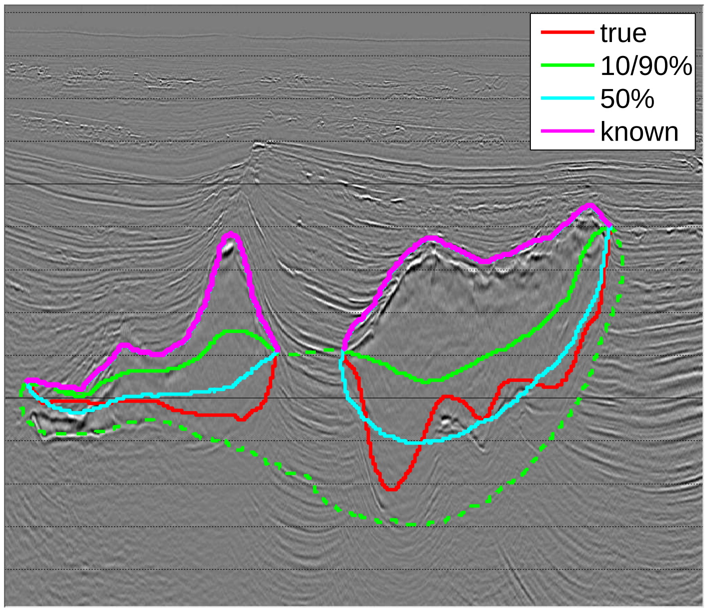




Samples generated with top salt fixed.

Real Geometry: Results Overlaid on Seismic



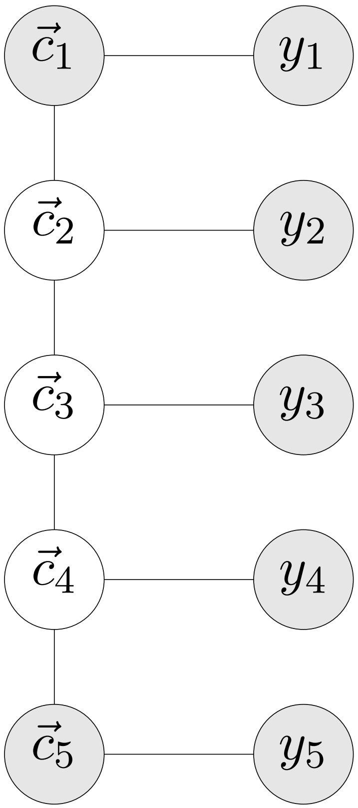


(in real life, this creates a registration problem)

Sampling Surfaces

- Extending our 2D curve sampling formulation to three dimensions is not straightforward, primarily because a canonical parameterization does not exist.
- Let $\vec{S} : [0,1] \times [0,1] \to \mathbb{R}^3$ be a surface and Y be the observed 3D volume.
- We construct a collection of curves on equally-spaced parallel slices $\mathcal{S} = \{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_N\}$.
- We then approximate the process of sampling from $\pi(\vec{S}|Y)$ by sampling from $\pi(\mathcal{S}|Y)$.

Hybrid 2D/3D Markov Model



- Construct target distribution as undirected Markov chain.
- This leads to the following factorization of $\pi(\mathcal{S}|Y)$ in terms of potential functions:

$$\prod_{i=1}^{N} \Phi_{\vec{c}_i}(\vec{c}_i) \prod_{i=1}^{N} \Phi_{\vec{c}_i, y_i}(\vec{c}_i, y_i) \prod_{i=1}^{N-1} \Phi_{\vec{c}_i, \vec{c}_{i+1}}(\vec{c}_i, \vec{c}_{i+1})$$

- $\Phi_{\vec{c}_i}(\vec{c}_i)$ and $\Phi_{\vec{c}_i, y_i}(\vec{c}_i, y_i)$ involve intra-slice interactions. $\Phi_{\vec{c}_i, \vec{c}_{i+1}}(\vec{c}_i, \vec{c}_{i+1})$ models inter-slice interactions (*e.g.*, dynamic shape models).

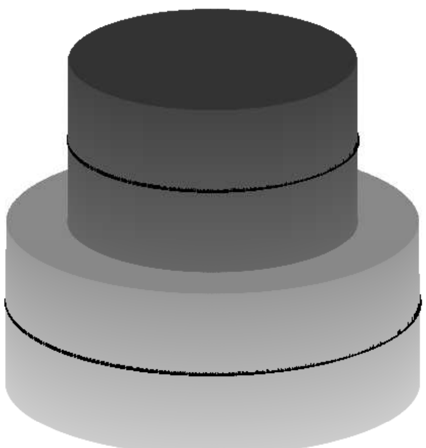
Slice-based Surface Area Model

- The standard regularizing term for 2D curve evolution is curve length. The analogous quantity in 3D is surface area.
- Consider a slice-based approximation to surface area:

$$\oiint_{\vec{S}} \mathrm{d}A \approx \iint_{\mathbf{0}\oplus\vec{c}_1} \mathrm{d}A + \sum_{i=1}^{N-1} \iint_{\vec{c}_i\oplus\vec{c}_{i+1}} \mathrm{d}A + \iint_{\vec{c}_N\oplus\mathbf{0}} \mathrm{d}A \ .$$

- Need to define a natural surface construction method to connect curves in adjoining slices (minimal surfaces are not geometrically accurate as they bow inwards).

Template Metric as a Coupling Term



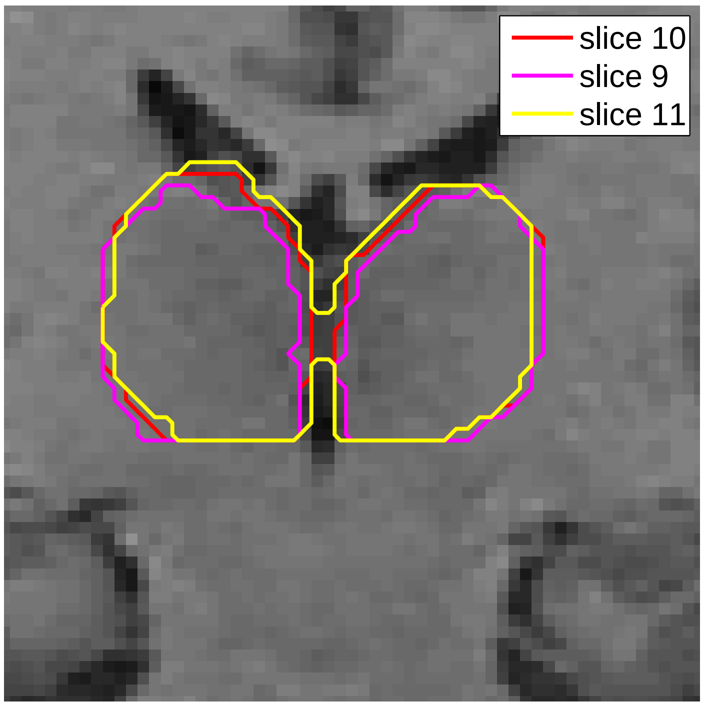
- If we approximate the coupling areas as piecewise-constant (in the z-direction), we get a stacked cylinder approximation.
- Define template metric (or symmetric area difference) as:

$$\mathrm{d}_{\mathrm{SAD}}(\vec{C}_1, \vec{C}_2) = \iint\limits_{\mathcal{R}(\vec{C}_1)} \mathrm{d}\boldsymbol{x} + \iint\limits_{\mathcal{R}(\vec{C}_2)} \mathrm{d}\boldsymbol{x} - 2 \iint\limits_{\mathcal{R}(\vec{C}_1)\cap\mathcal{R}(\vec{C}_2)} \mathrm{d}\boldsymbol{x} \ .$$

- We can write the slice-coupling surface area as:

$$\iint\limits_{\vec{c}_i \oplus \vec{c}_{i+1}} \mathrm{d}A = \frac{\Delta z}{2} \oint_{\vec{c}_i} \mathrm{d}s + \frac{\Delta z}{2} \oint_{\vec{c}_{i+1}} \mathrm{d}s + \mathrm{d}_{\mathrm{SAD}}(\vec{c}_i, \vec{c}_{i+1})$$

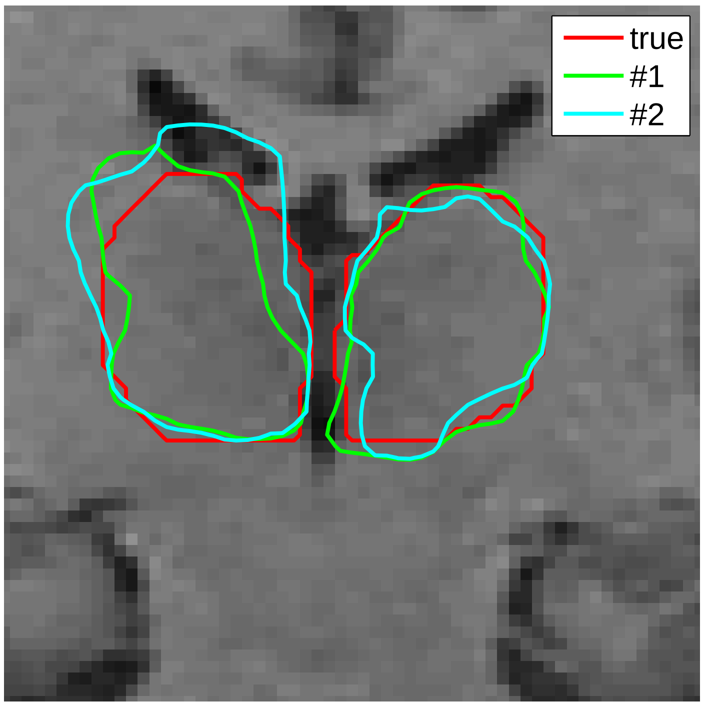
Neighbor Slice Constraints

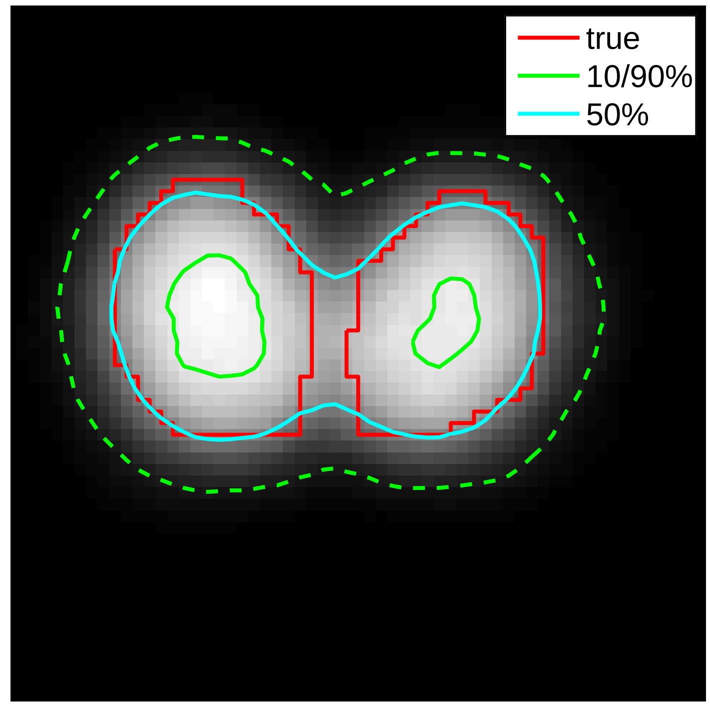


- Consider the situation where we are given $\vec{c}_{n-1}$ and $\vec{c}_{n+1}$ and we wish to $\vec{c}_n$.
- Due to Markov nature of model, $\vec{c}_i$ is conditionally independent of all other slices, so we simply have a 2D curve sampling problem with additional terms in the energy (can view as shape priors):

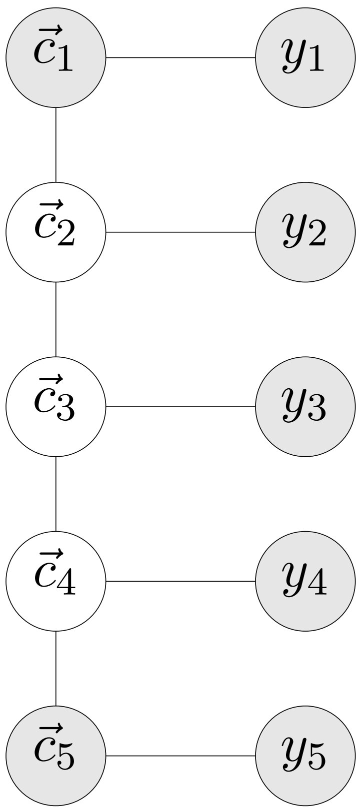
$$\pi(\vec{c}_n \,|\, \mathcal{S} \setminus \vec{c}_n, Y) \propto \Phi_{\vec{c}_n}(\vec{c}_n)\Phi_{\vec{c}_n, y_n}(\vec{c}_n, y_n)$$
$$\Phi_{\vec{c}_{i-1}, \vec{c}_i}(\vec{c}_{i-1}, \vec{c}_i)\Phi_{\vec{c}_i, \vec{c}_{i+1}}(\vec{c}_i, \vec{c}_{i+1})$$

Thalamus: Neighbor Slice Results



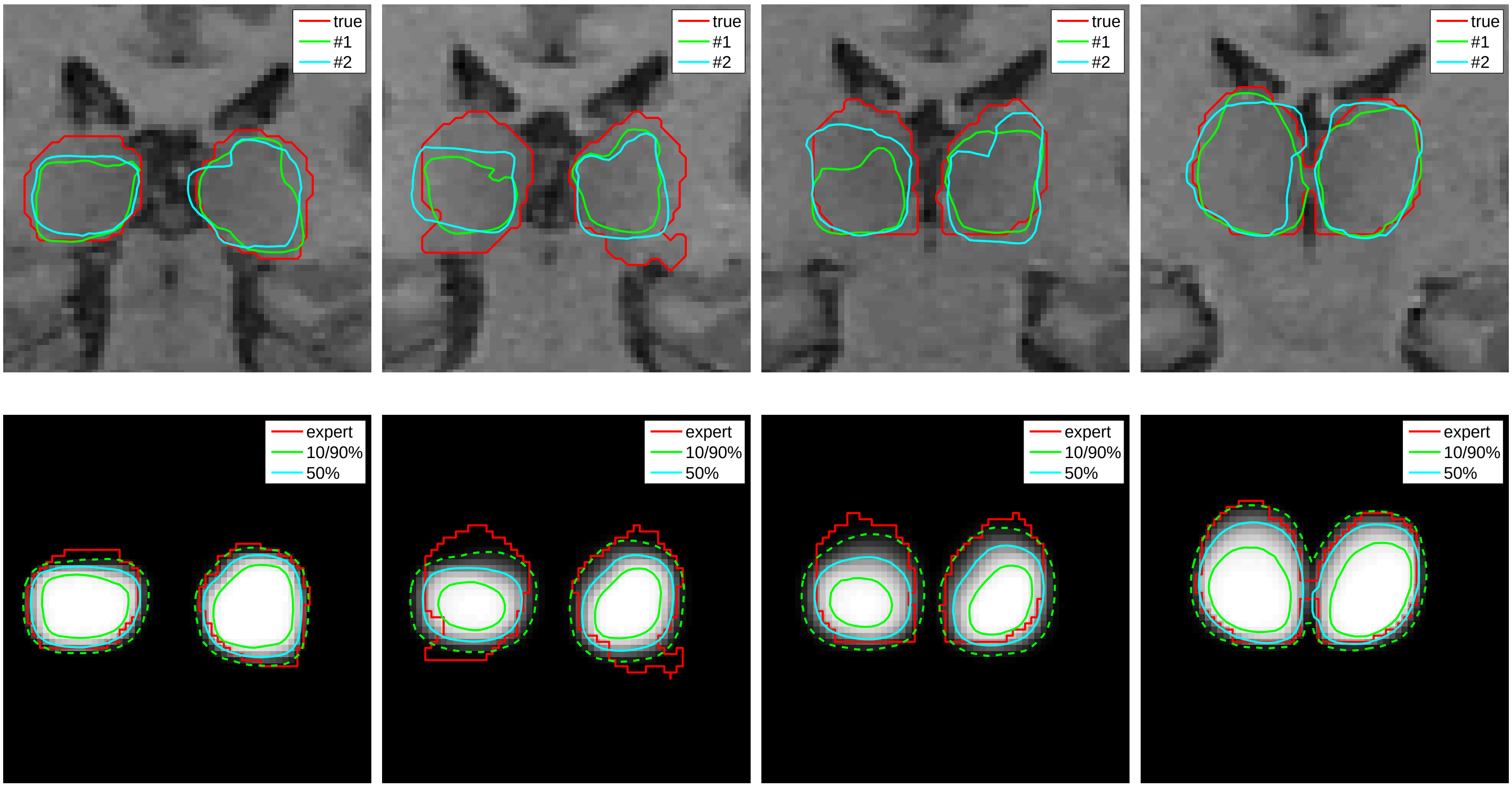


Gibbs Sampling for Multiple Slices

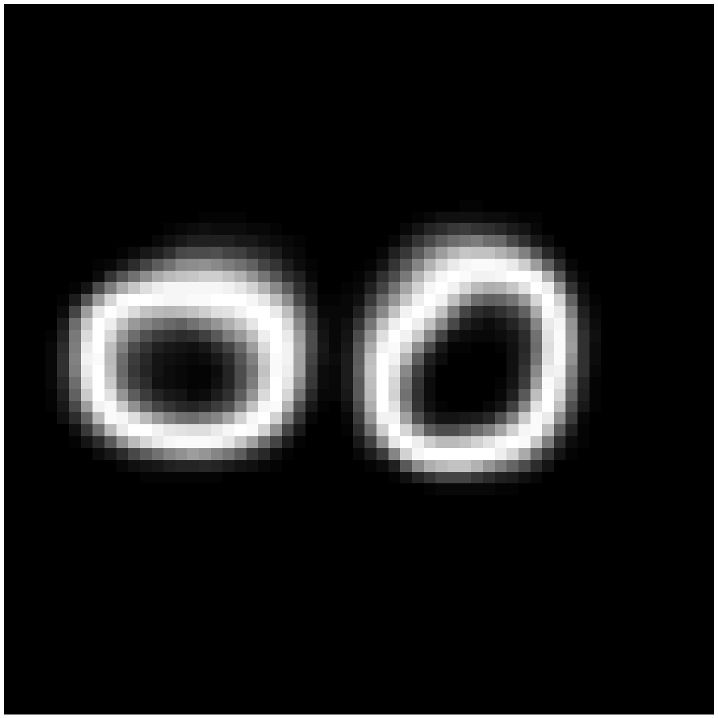


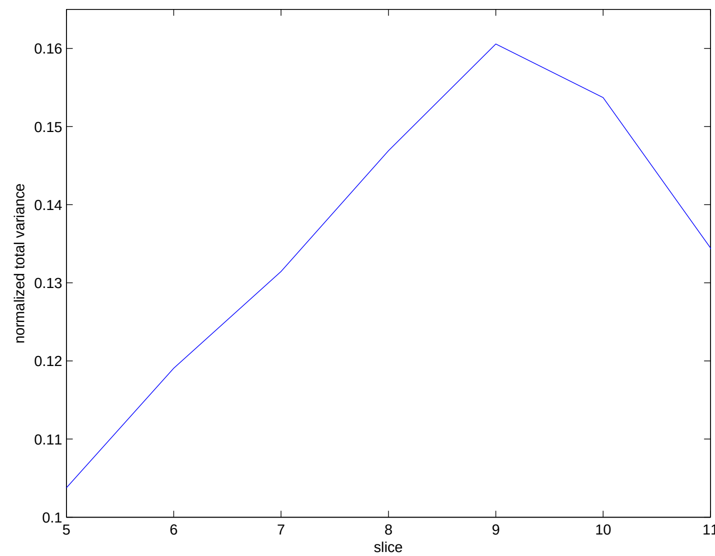
- If there are contiguous unknown slices, we need to be able to sample from the joint distribution over those slices (discontiguous groups of slices may still be processed independently).
- One option is to do Gibbs sampling and iteratively sample from $\mathrm{p}(\vec{c}_i|\vec{c}_{i-1}, \vec{c}_{i+1}, y_i)$. i can be changed randomly or deterministically.
- We do not know how to sample from p directly. Instead, we can do N Metropolis-Hastings steps (same formulation as for the 2D case) and still have detailed balance hold.

Thalamus: Slices 5, 7, 9, and 11



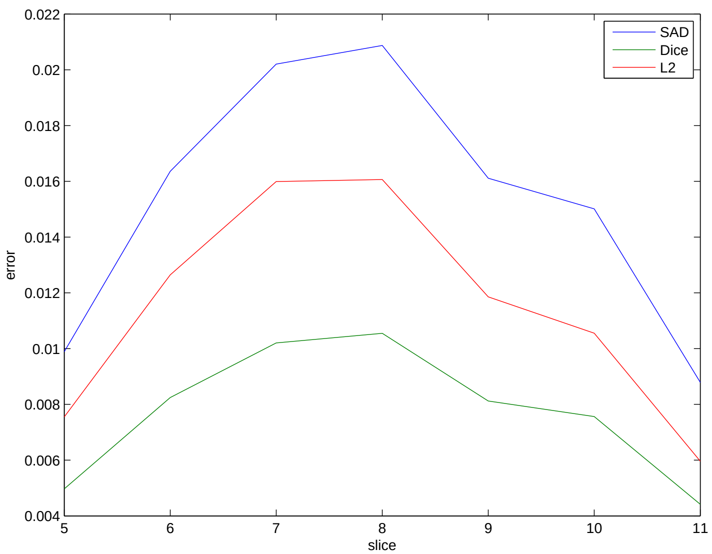
Variability Per Slice





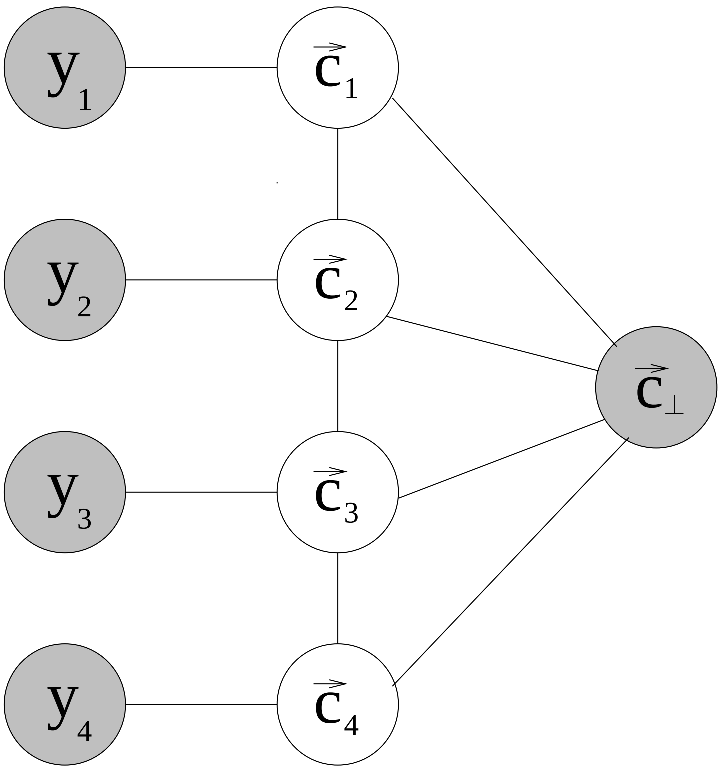
On left, variance map from histogram image ($\sigma^2 = p(1-p)$).
On right, sum of variance per slice normalized by area of true curve.

Error Per Slice



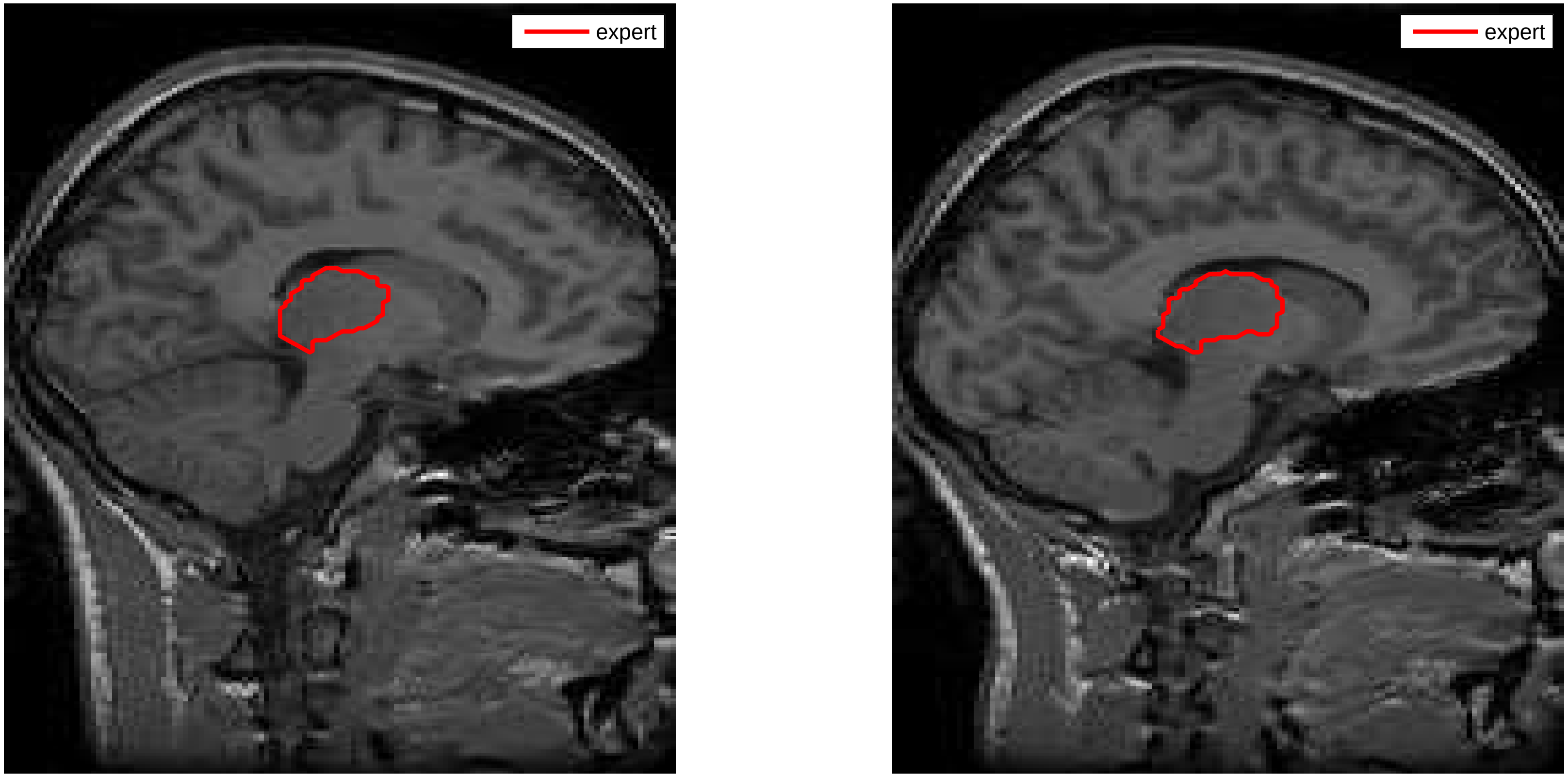
- We can use a number of methods to compare our sampling results with the expert segmentations:
  - Symmetric area difference (SAD) between median contour and expert contour.
  - Dice measure (weights correct labels twice as much as incorrect labels) between median and expert contours.
  - L2 distance between histogram image and binary 0/1 expert label map.

Orthogonal Slice Information

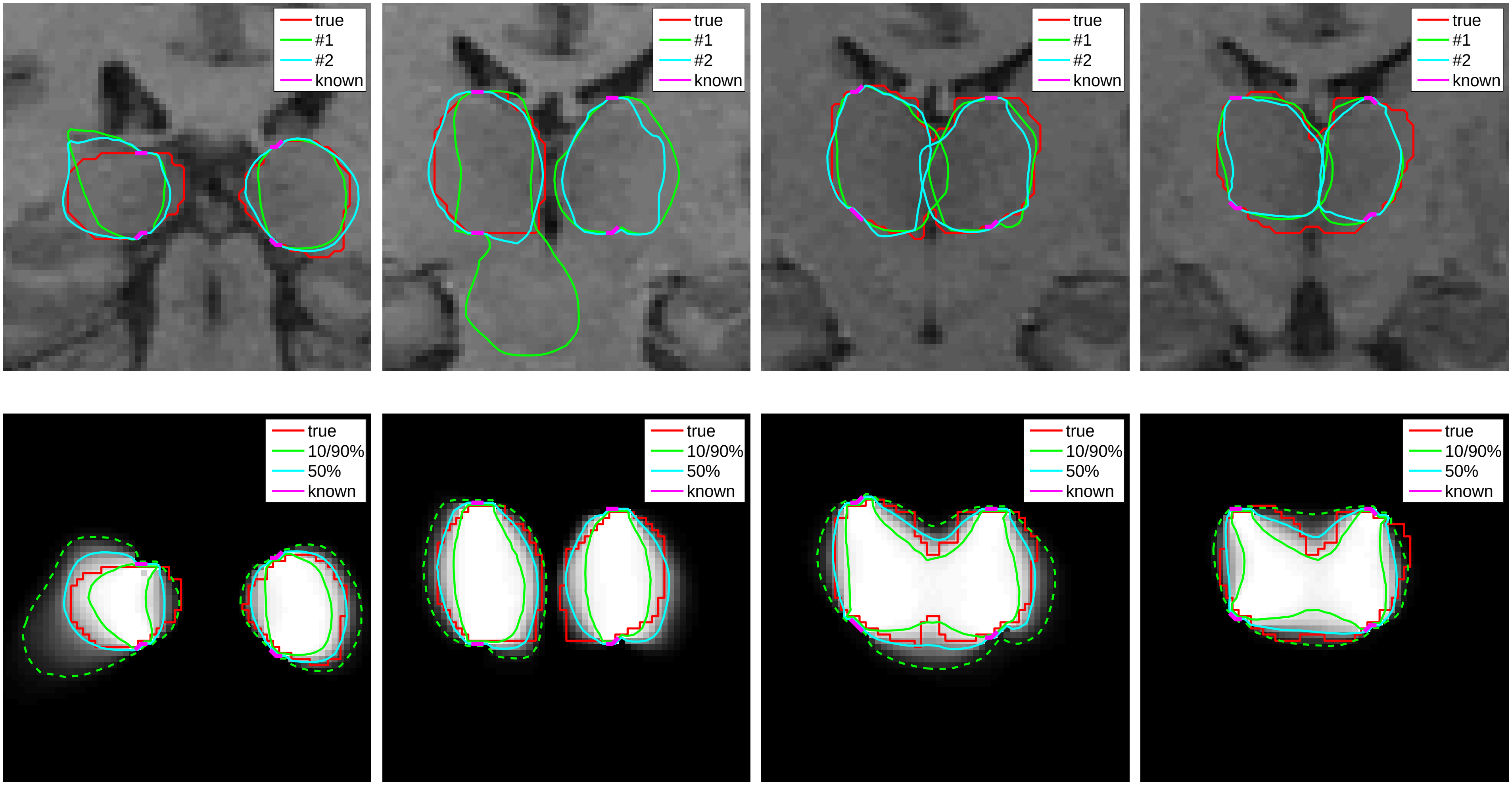


- An alternative model is to consider having slices oriented in orthogonal directions.
- If $\vec{c}_\perp$ is known and fixed, this is equivalent to fixing the segmentation values along a row or column in the original slices.
- Thus we can incorporate this information simply using our 2D conditional sampling framework.

Sagittal Expert Segmentations



Sagittal Constraints, Axial Slices



Conclusions and future work

Summary

- Sampling provides a number of benefits over standard optimization-based curve evolution techniques (*e.g.*, robustness to local minima).
- We extended our original formulation to allow conditional simulation for semi-automatic segmentation applications.
- We also constructed a hybrid 2D/3D framework to sample surfaces.

Future work

- Incorporate uncertainty into user information.
- Orthogonal 2.5D chains which allow conflicting state information.