

***MACHINE LEARNED BOUNDARY  
DEFINITIONS...***

***The True Story of A Ten-Year Trail  
Across the Ph.D. Plains***

Stewart Crawford-Hines

BMAC -- 27 October 2003

# Outline

*Results* (10)

*Comparisons* (7, 10)

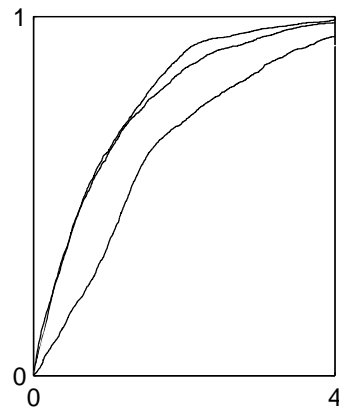
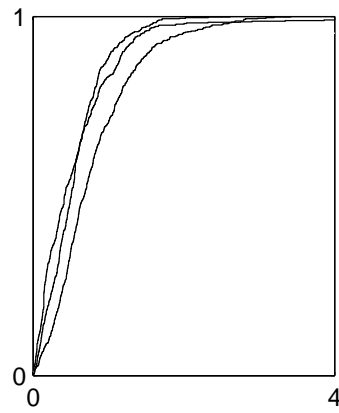
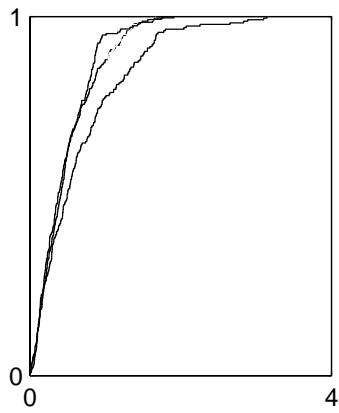
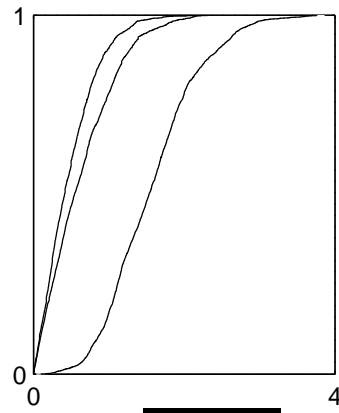
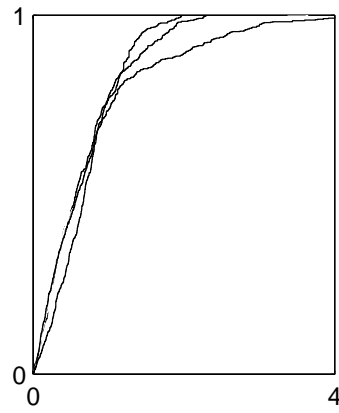
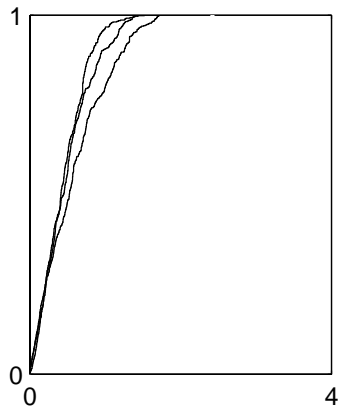
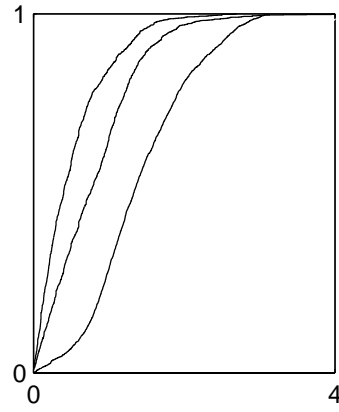
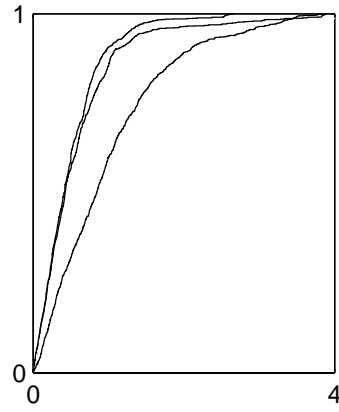
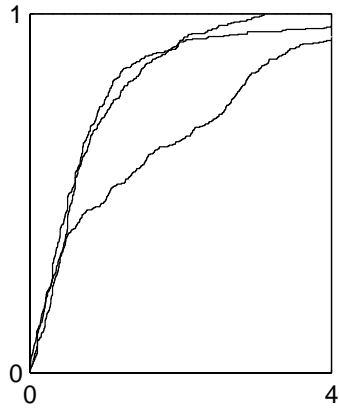
*Quantification* (6)

*Engineering* (2-8)

*Inspiration* (1)

*Funders and Helpers*

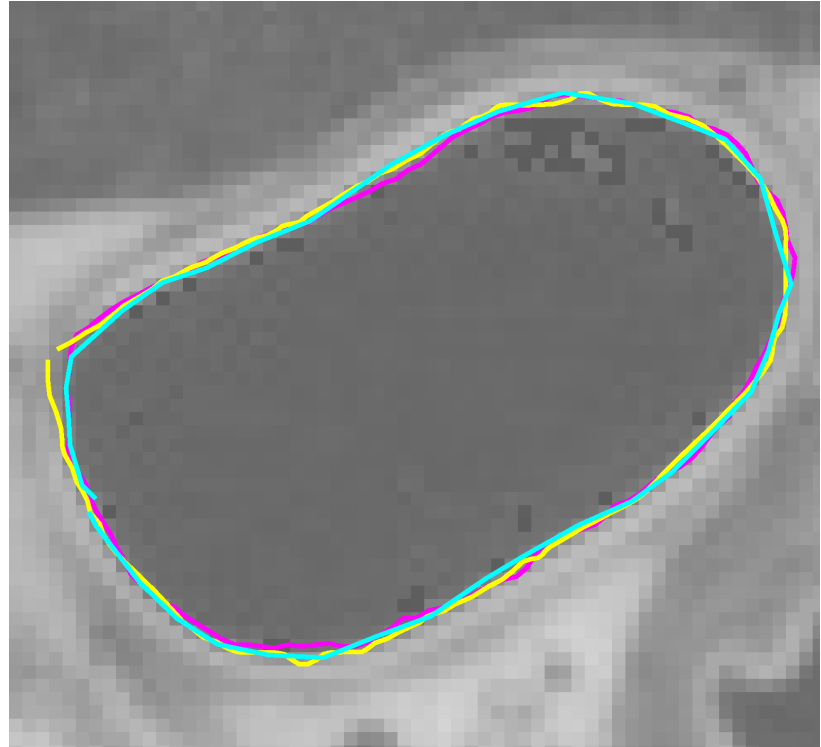
# ***Results*** (10)



# Imagery

- “Easy” / Straightforward to automate:

Many existing methods and techniques work well



- “Difficult” / Impossible to automate:

Nothing works; requires domain expertise for adequate segmentation





# Summary

- In the middle ground between “easy” and “hard images”, boundaries in the world are not always represented by edges in the image. Some of these boundaries will have patterns that can be learned.
- Expert’s Tracing Assistant (ETA) developed to validate this premise.
- ETA performs comparably on “easy” images, does better where *a priori* edge / boundary definitions fail.
- Domain: Large, repetitive image sets

# ***Comparisons*** (7, 9)



# Comparison to User-Guided, State-of-the-Art Methods

For a set of representative structures from the Visible Human imagery,  
compare the performance and application of ETA to:

Intelligent Scissors (IS)

Active Contour Models (ACM)

**Performance Measure:** compare the **best** boundaries these methods produce to a Ground Truth, and judge the difference with respect to natural intra-expert variation.

# Comparisons ???

(1) Bring their methodology to your domain...

- who writes / runs the software ???
- Kevin Bowyer's posed dilemma (*in an earlier BMAC*)

2) Bring your domain to their methodology...

# Active Contour Models

- An ACM is an energy minimizing spline.
- Model is initialized close to a structure of interest and then iterated into an energy minima.
- An energy function of the boundary contour is defined so that minima correspond to boundaries of interest in images. Two main components:
  - The shape component: first and second derivatives of curvature
  - The image component: defined on the image plane such that local minima correspond to edges (e.g., an inverted intensity gradient)
- Transformed to a dynamic system for numerical solution.
- **Parameters  $\alpha$  and  $\beta$  weight the shape component contribution (“*tension*” and “*rigidity*”); parameters  $\gamma$  and  $\kappa$  weight the damping and inertial forces of the dynamic system.**





# Intelligent Scissors

- In an initial preprocessing step, a local cost from every pixel to its eight neighbors is pre-computed.
  - The cost function is a linear combination of 1<sup>st</sup> and 2<sup>nd</sup> derivative measures across image intensity, plus possibly some local statistics.
- The image is viewed as a weighted graph, with pixels as nodes, where each pixel has weights on the eight graph edges to its neighbors.
- The user manually places a starting point on the boundary of interest.
- The system follows a minimal cost path from most recent control point to the cursor's current screen location.
- As the cursor moves, this path is updated in real time and appears to be a wire snapping around on the edges in an image.
- Final boundary, defined on pixel centers, is smoothed with a weighted average.



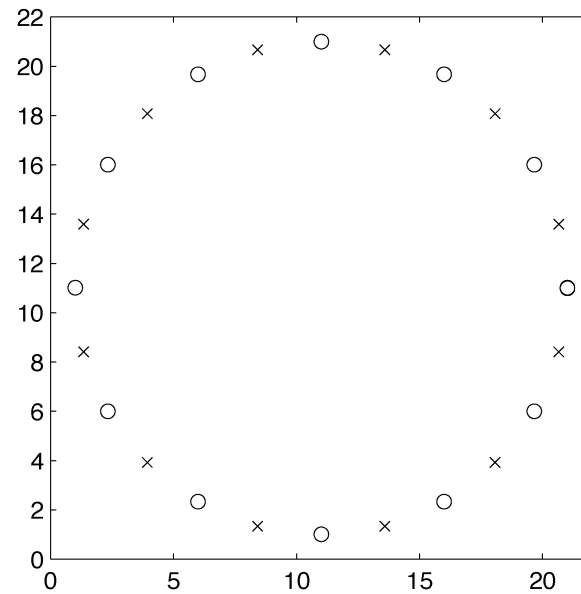




# ***Quantification*** (6)

# Quantifying Boundary Differences

I'm working with boundaries, but only have points:

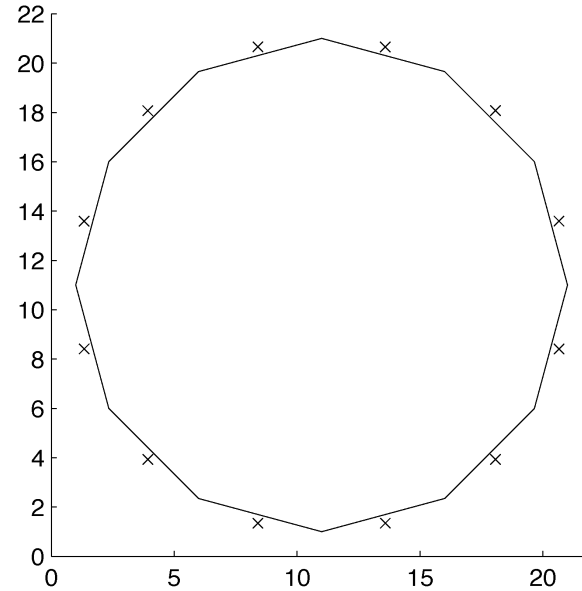
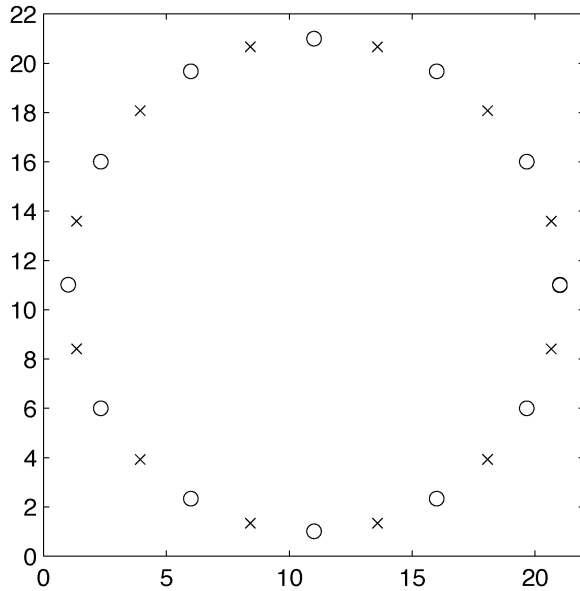


## Side-trips in Measurement Meadow

- Initial measurements look at distance from one point to closest 'other curve' point
- Reviewers ask about their favorite measures, e.g., a Hausdorff distance
- Literature is full of other ideas
- Finally - Define a measure appropriate to the domain at hand

## Quantifying Boundary Differences - 2

The Problem: Want differences between boundaries, but only have points.

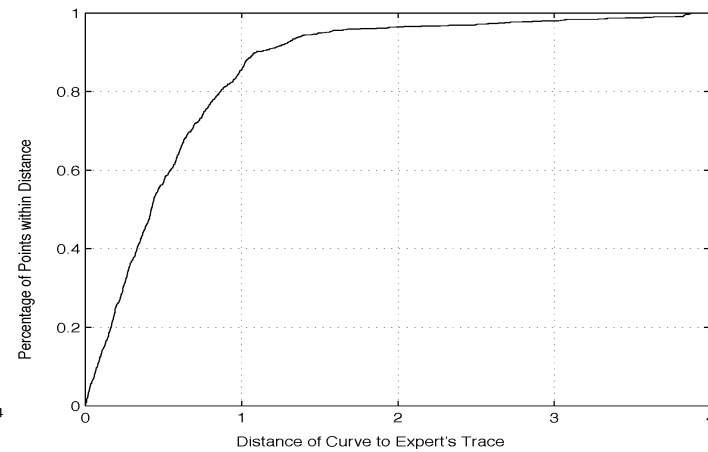
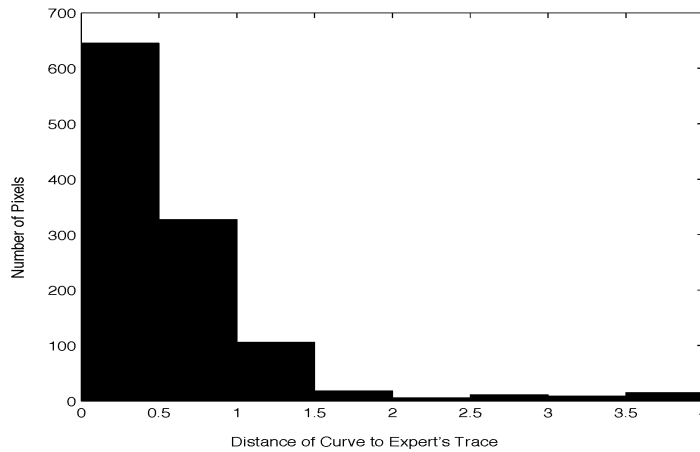
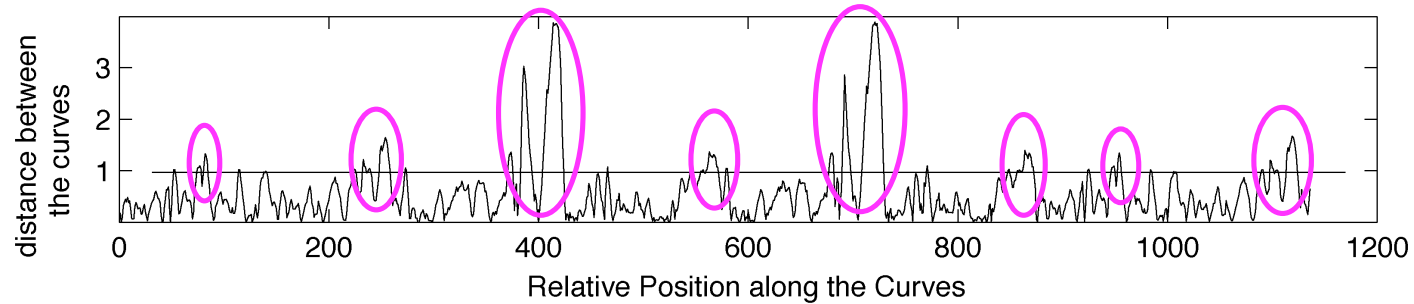


LEFT: Two sets of points, marked with **x** and **o**, taken from a circle of radius 10 centered at (11,11).

RIGHT: Measuring from one set of points to the polyline of the other set.

# Using Distance Sets

Set of Boundary Differences (to, then from, GT) compared.



Eight excursions noted above a 1.0 threshold imply 4 user corrections needed.







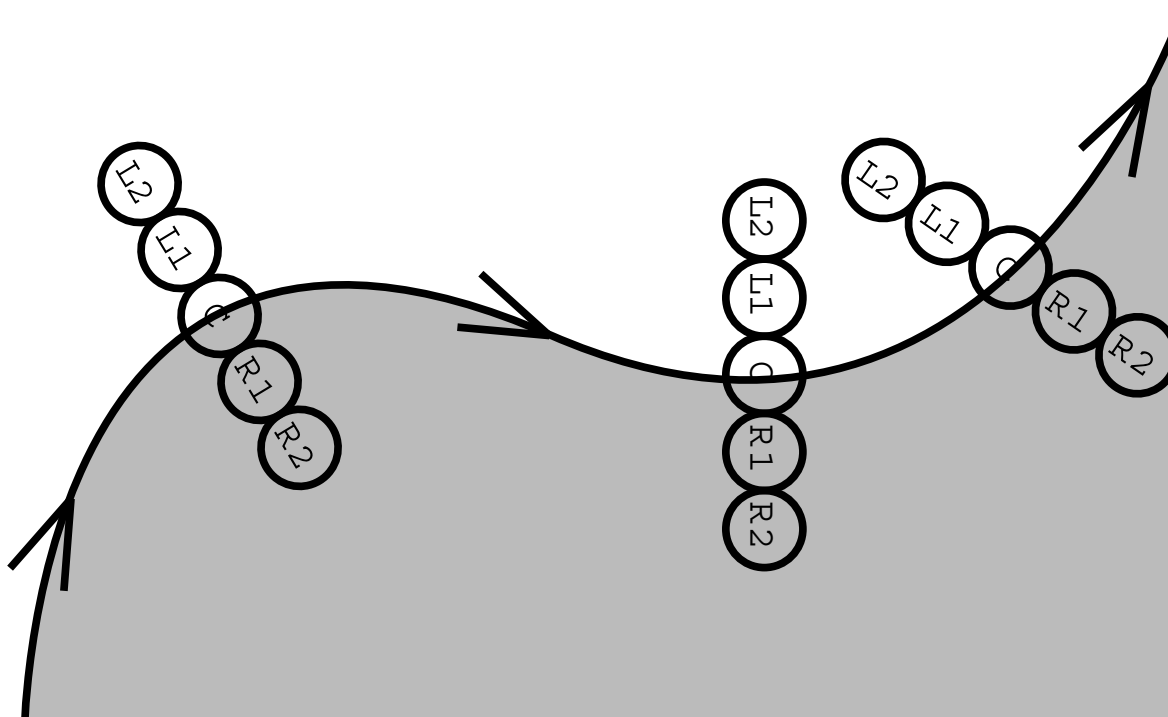


		<u>% of Curve within 1 pixel of GT</u>				<u>90<sup>th</sup> percentile of distance to GT</u>			
		M2T	ETA	ACM	IS	M2T	ETA	ACM	IS
<b>B A S I C</b>	Leg Bone	80%	96%	91%	82%	1.25	.84	.95	1.42
	Throat	95%	88%	78%	86%	.87	1.06	1.52	1.07
	Leg Muscle	73%	77%	77%	70%	2.10	1.27	1.48	1.52
	Ventri- cle	89%	84%	71%	70%	1.03	1.18	1.54	2.35
<b>BASIC average</b>		<b>84%</b>	<b>86%</b>	<b>79%</b>	<b>77%</b>	<b>1.31</b>	<b>1.09</b>	<b>1.37</b>	<b>1.59</b>

<b>I N T</b>	Leg Skin	78%	91%	18%	13%	1.27	.97	2.43	2.74
	Arm Skin	84%	65%	29%	27%	1.21	1.52	2.40	2.62
<b>INT average</b>		<b>81%</b>	<b>78%</b>	<b>24%</b>	<b>20%</b>	<b>1.24</b>	<b>1.25</b>	<b>2.42</b>	<b>2.68</b>
<b>H A R D</b>	Arm Bone	75%	71%	48%	53%	1.89	1.86	3.45	3.27
	Arm Muscle	86%	91%	60%	73%	1.09	.97	2.09	1.67
	Thorax Lobe	62%	62%	37%	60%	2.39	2.00	3.41	3.69
<b>HARD average</b>		<b>74%</b>	<b>75%</b>	<b>48%</b>	<b>62%</b>	<b>1.79</b>	<b>1.61</b>	<b>2.98</b>	<b>2.88</b>

***Engineering*** (2-8)

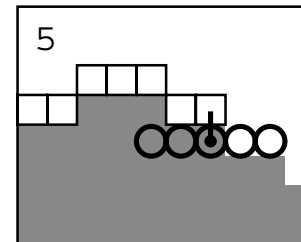
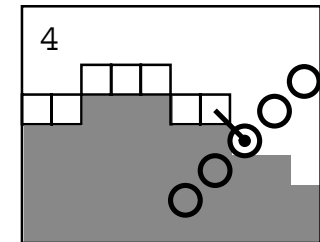
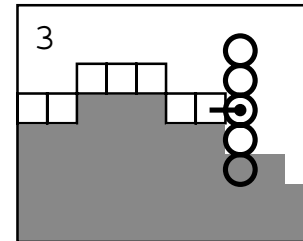
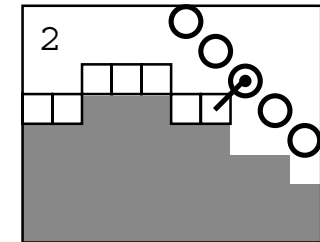
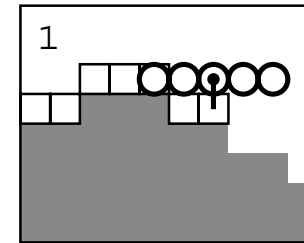
# Sampling Neighborhoods Along An Example Boundary



Boundaries separate things. **C** represents the center of a boundary neighborhood. **L** and **R** indicate immediate left and right neighbors of **C**.

# Creating Positive and Negative Exemplars

case	--- data ---					- response -
	L2	L1	C	R1	R2	on boundary ???
1	1.0	1.0	1.0	1.0	1.0	false
2	1.0	1.0	1.0	1.0	1.0	false
3	1.0	1.0	1.0	1.0	0.5	false
4	1.0	1.0	1.0	0.5	0.5	TRUE
5	1.0	1.0	0.5	0.5	0.5	false

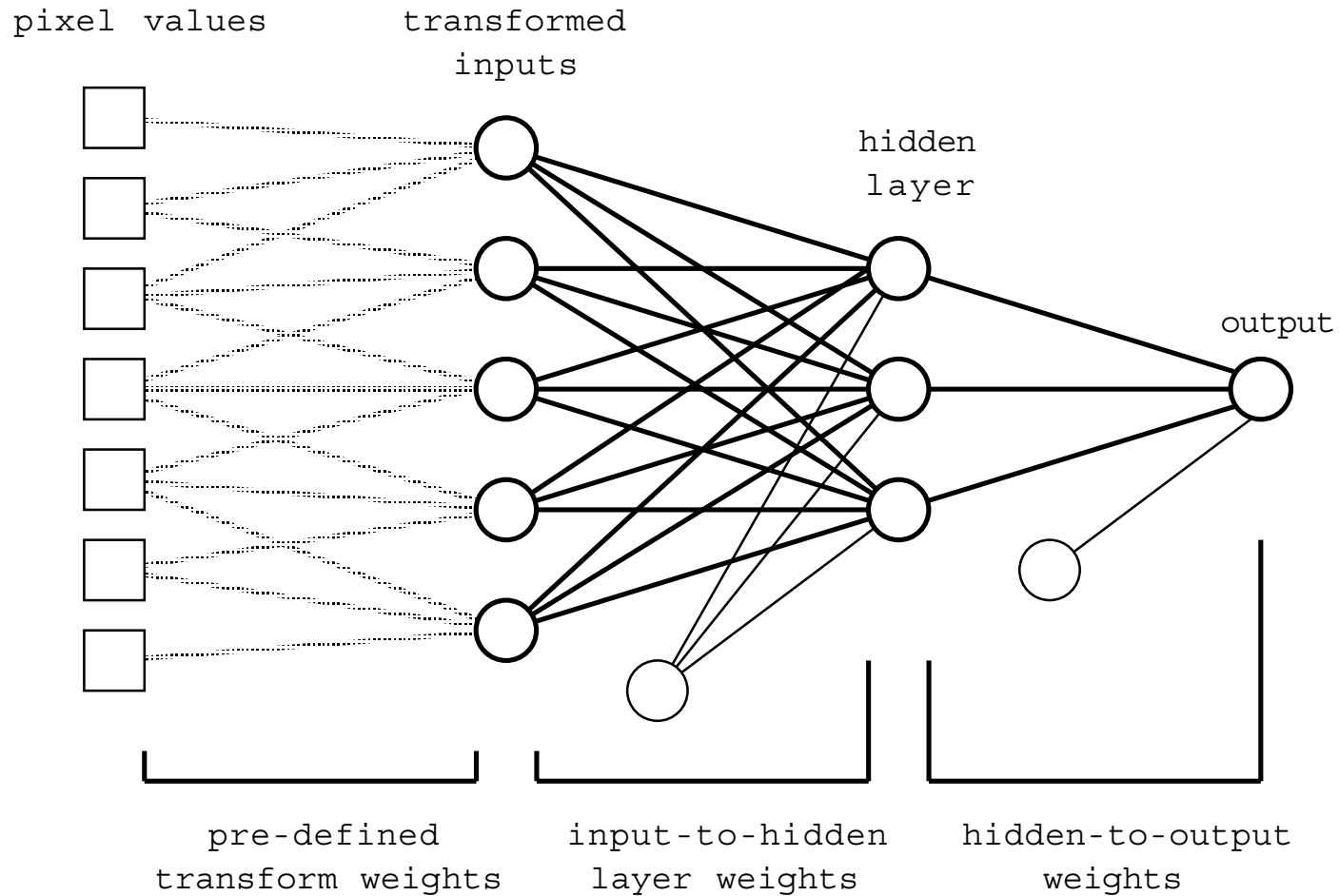


# Interpreting the Output

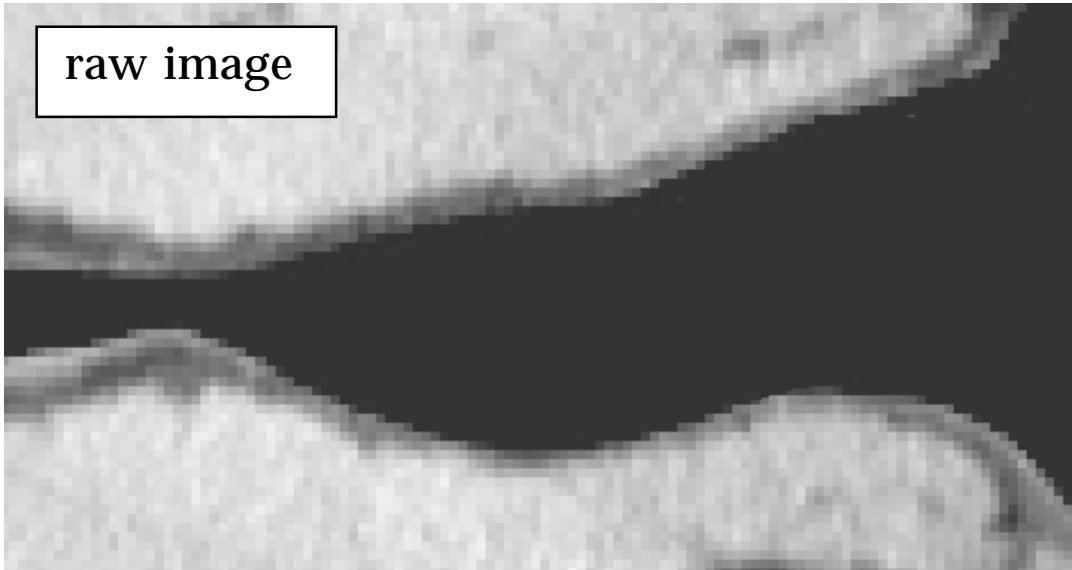
case	--- data ---					--- response ---		
	L2	L1	C	R1	R2	boundary?	Smooth Evalutaion Function (SEF)	Feature Detector (FD)
1	1.0	1.0	1.0	1.0	1.0	far left	0.1	0.1
2	1.0	1.0	1.0	1.0	1.0	far left	0.1	0.1
3	1.0	1.0	1.0	1.0	0.5	near left	0.3	0.1
4	1.0	1.0	1.0	0.5	0.5	YES	0.5	0.9
5	1.0	1.0	0.5	0.5	0.5	near right	0.7	0.1



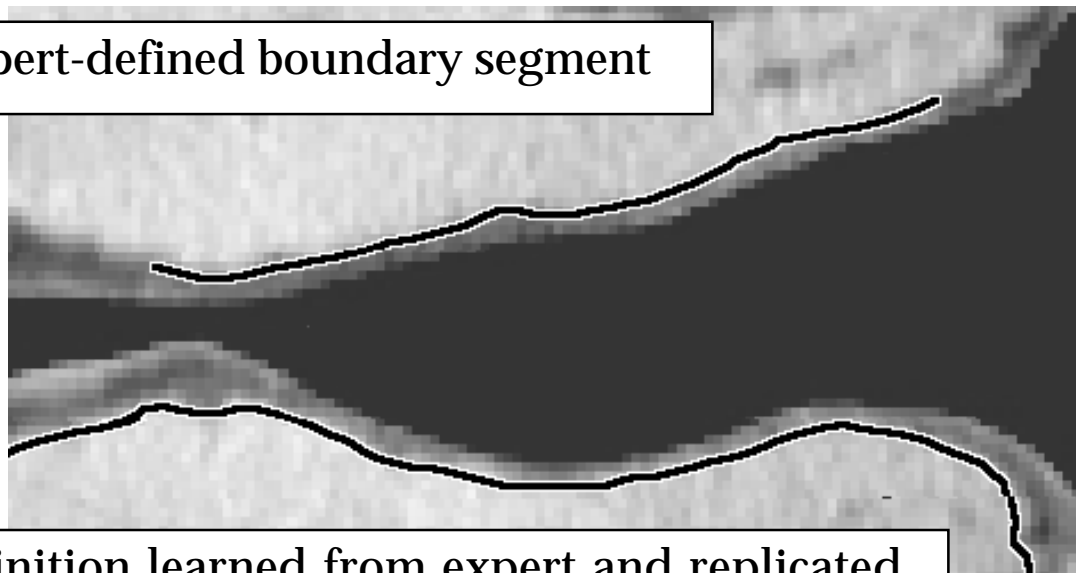
# Filtering the Inputs Effectively Adds a Layer



raw image



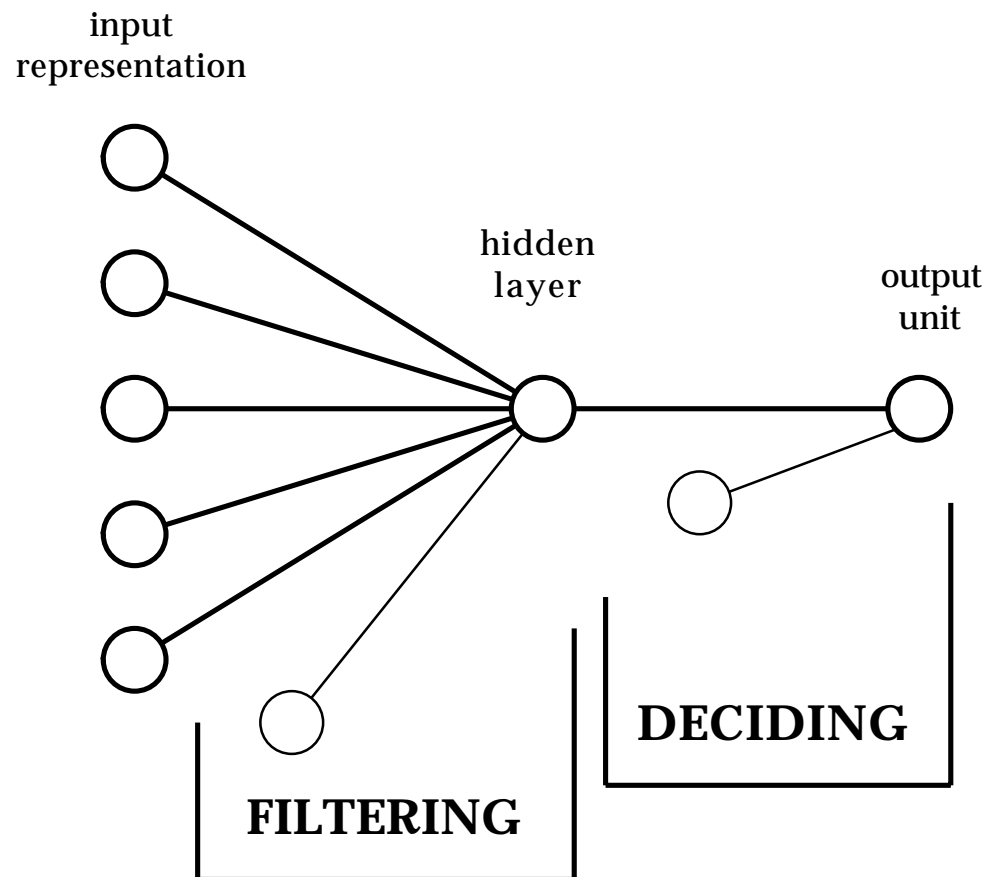
expert-defined boundary segment



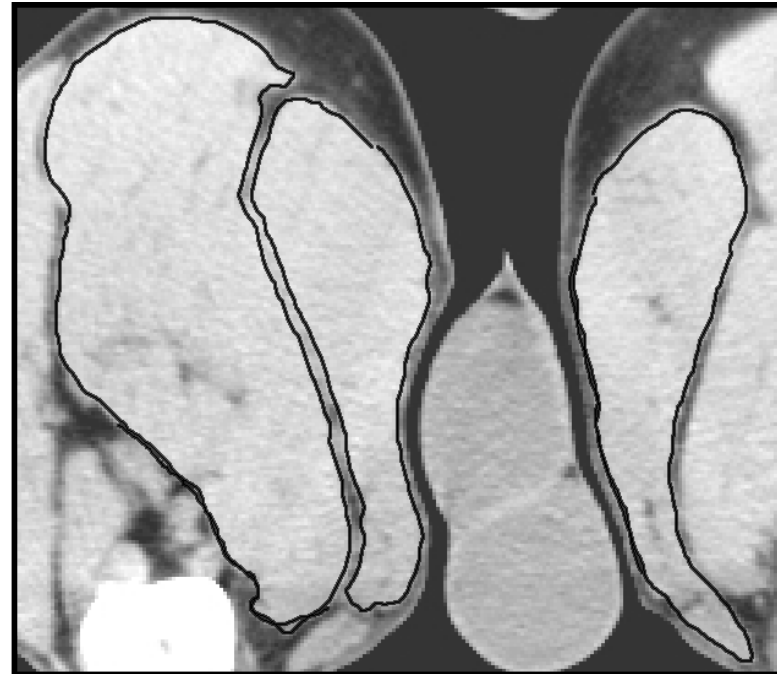
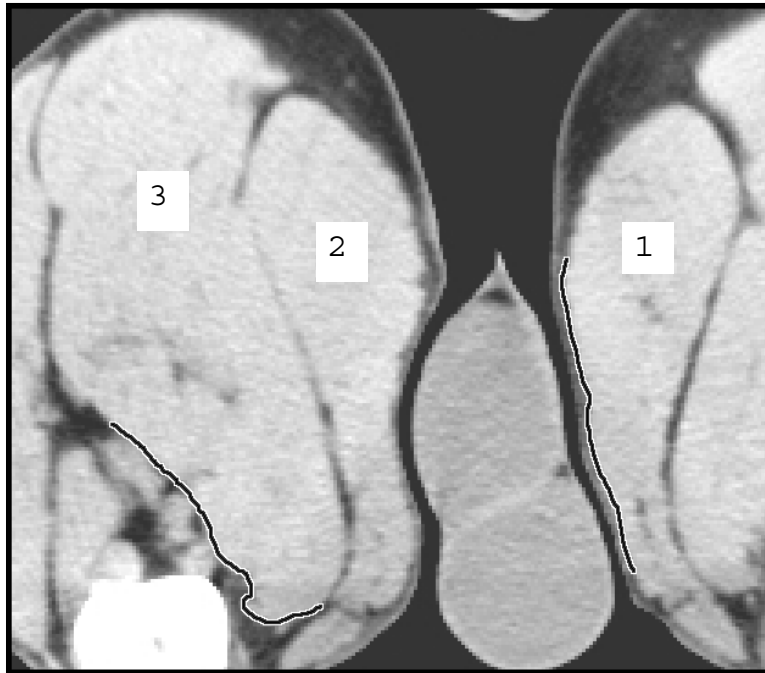
definition learned from expert and replicated



# Interpreting the Neural Net's Learning



## Application in CT images



***Inspiration*** (1)

**BMAC – Oct/Nov 1993**

The Versalius Project

The Challenge

# The *A Priori* Approach

This approach begins with assumptions, works through mathematics, then applies it to images.

- assume boundaries of interest are edges in image
- assume some definition of an edge, typically: the locally strongest discontinuity in image intensity
- relate definition to mathematics of intensity function: 1<sup>st</sup> derivative extrema, 2<sup>nd</sup> derivative zero-crossing, phase shift in complex plane, ...
- define a filter to implement the transform
- derive criteria for filtered images to select relevant edges

## The *Learn What's Needed* Approach

- On a (small) subset of imagery, expert defines segments of the boundary of interest in representative areas.
- Segments used to create a set of positive and negative exemplars.
- From exemplars, supervised learning method learns the pattern that characterizes this boundary of interest. (A confidence measure, also?)
- In new images, expert identifies a start for a boundary and the system traces ahead automatically. (Continues while confident.)

# Learned Boundaries - Notes

- Learned boundary can be only as good as the representative sample.
- Expert is in the process as a monitor; provides corrections to the system when it errs or when confidence measure is low.
- Learning is only effective in large, repetitive image sets, where the cost of learning (over some small imagery subset) can be recouped by automatically processing the remainder of the imagery.
- Note the flow is opposite of the *a priori* approach:

Start with imagery, define a learning method, then finally derive and apply boundary definitions

- Many possibilities exist for extension and learning methodologies. Goal is to implement one to prove the premise. No claims of optimality implied.

# Objectives

**Problem:** Boundary and Edge assumptions limit the range of applicability of these *a priori* methods.

**Proposal:** Start without boundary assumptions, learn the boundary definition to match the boundary of interest.

**Method:**

- Develop a framework for boundary learning and tracing.
- Verify its adequacy on sample imagery
- Compare it to other state of the art user-guided methods
- Evaluate methods in comparison to tracing skills of experts  
*“The boundary is within X pixels of the expert Y% of the time”*
- Identify limitations of framework and avenues of future work

# ***Funders & Helpers***

# Funding

Year	money aligned with research	money elsewhere
1	<i>(wrote 1<sup>st</sup> software prototype, demo'd to Versalius team)</i>	RA – Reinforcement Learning
2-4	<i>(PhD proposal; wrote 2<sup>nd</sup> prototype to validate basic premise)</i>	TA – Graphics <i>corporate T&amp;E</i>
5	prototype used to secure CASI grant, joint CSU-VP funding <i>(RA)</i>	<i>corporate T&amp;E</i>
6	CASI results used to secure NSF-I grant, VP funding <i>(contractor)</i>	
7-8	NSF-I work used to secure NSF-II grant, VP funding <i>(contractor)</i>	
9	<i>(none)</i>	1 <sup>st</sup> half - VP misc grants; 2 <sup>nd</sup> half - none
10	<i>(none)</i>	VP contract work for biomedical visualizations

# Helpers

- \* our faculty, within and outside the department
- \* conference reviewers
- \* conference attendees
- \* other universities (*e.g., BYU visit*) and researchers (*e.g., Terzopoulos*)