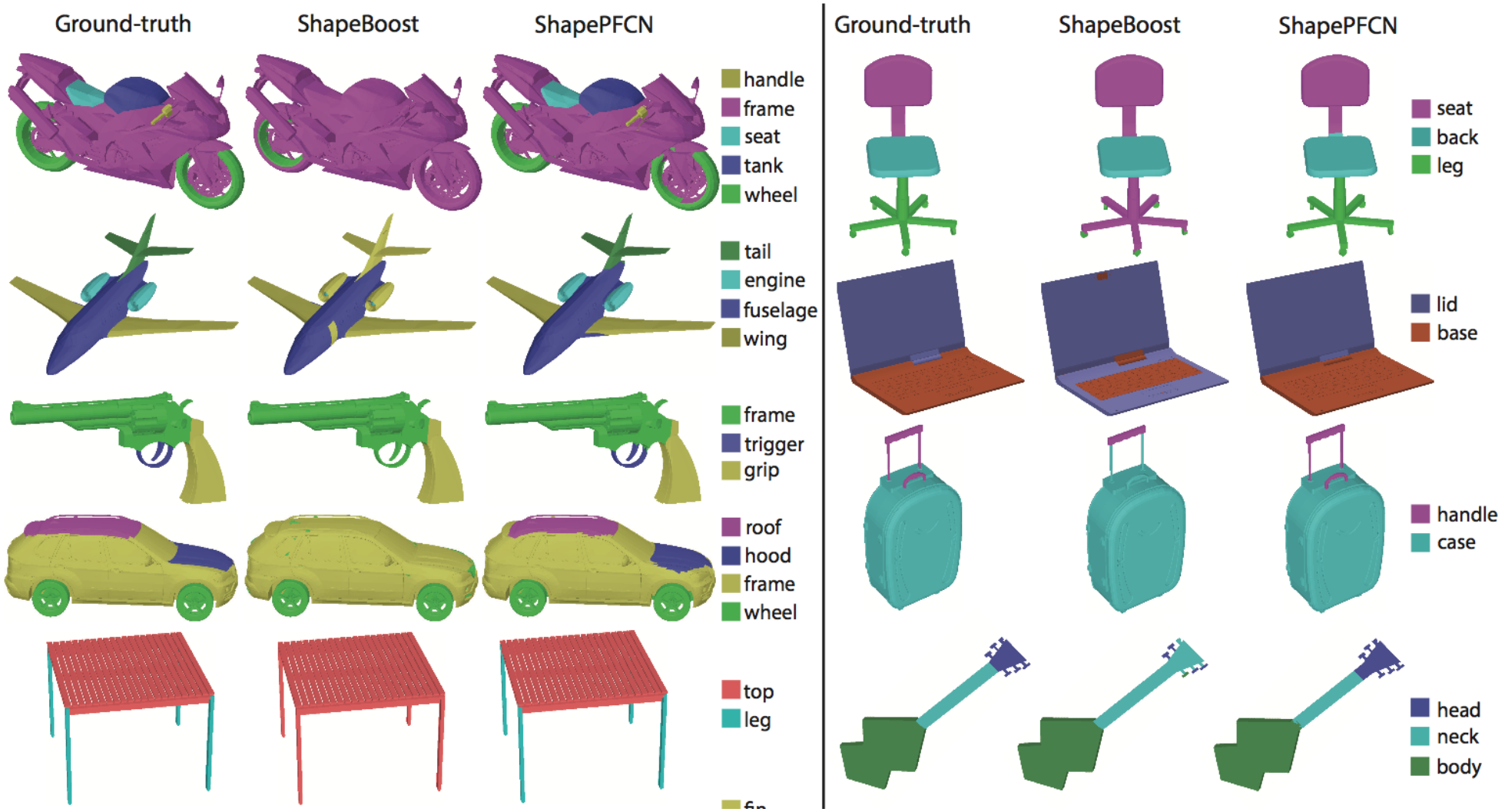


Supervised Learning for Shape Segmentation

Siddhartha Chaudhuri, CS344 guest lecture

(thanks to Vangelis Kalogerakis for many slides)

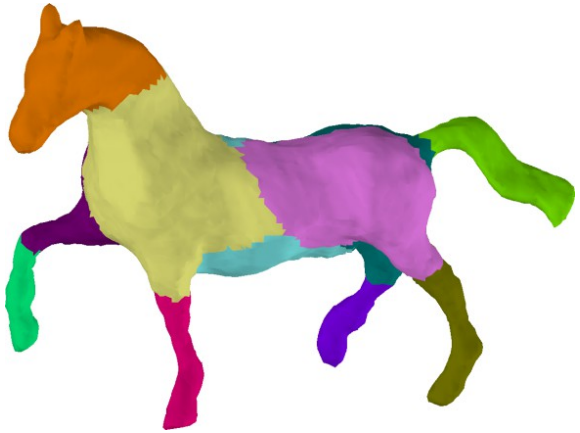
Problem Statement

- **Decompose** shape into structurally/semantically meaningful parts and **label** the components with meaningful names, e.g. “leg”, “ear”, “torso”

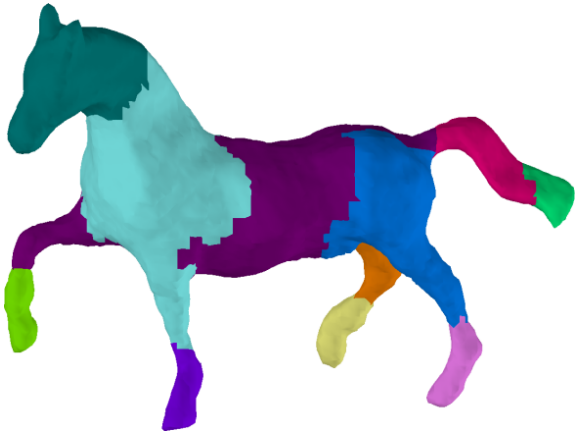
Not as easy as it looks...

Shape Diameter
[Shapira et al. 10]

Randomized Cuts
[Golovinskiy and Funkhouser 08]

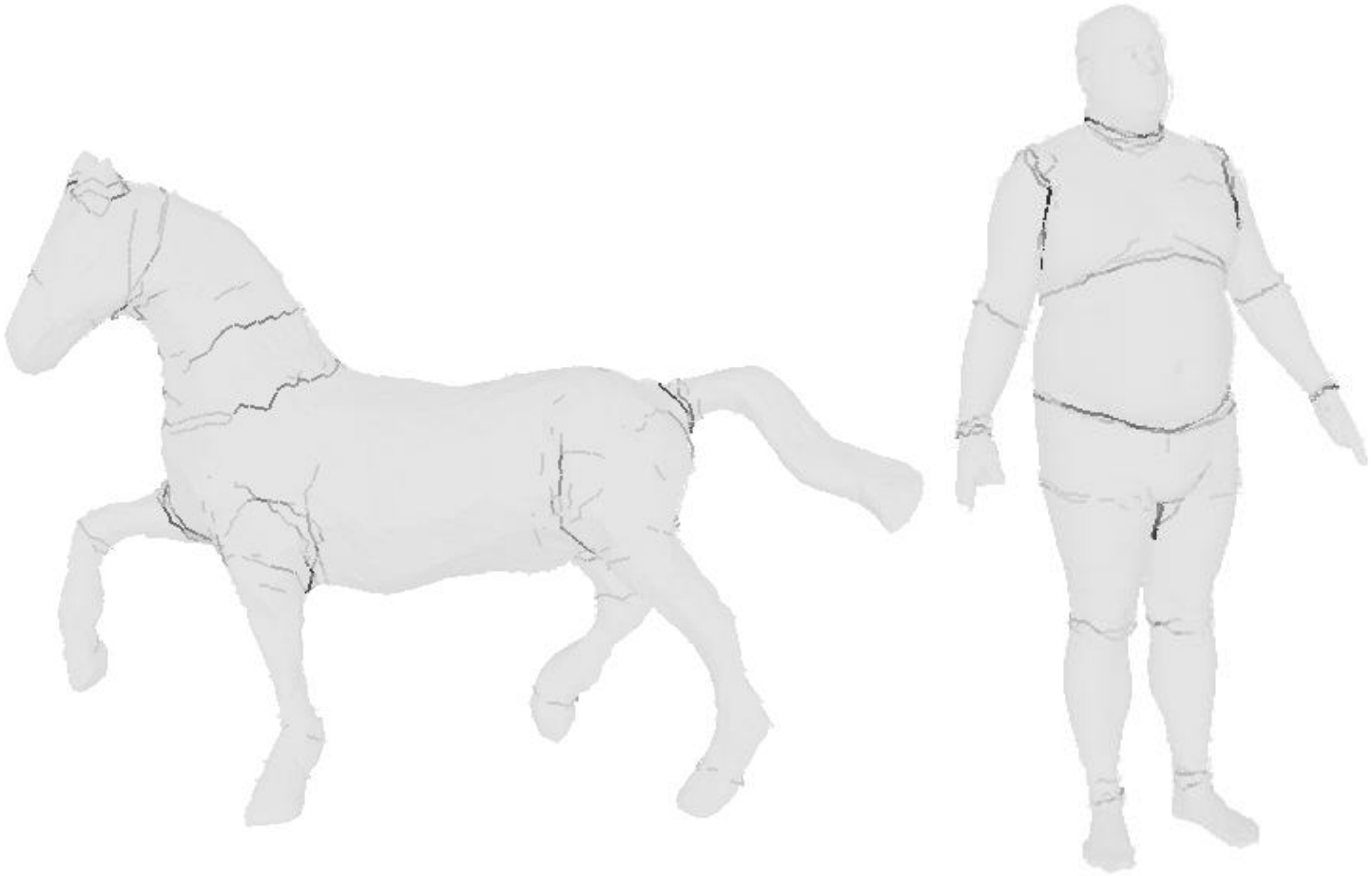


Random Walks
[Lai et al. 08]



Normalized Cuts
[Golovinskiy and Funkhouser 08]

Is human-level segmentation even possible without higher-level cues?



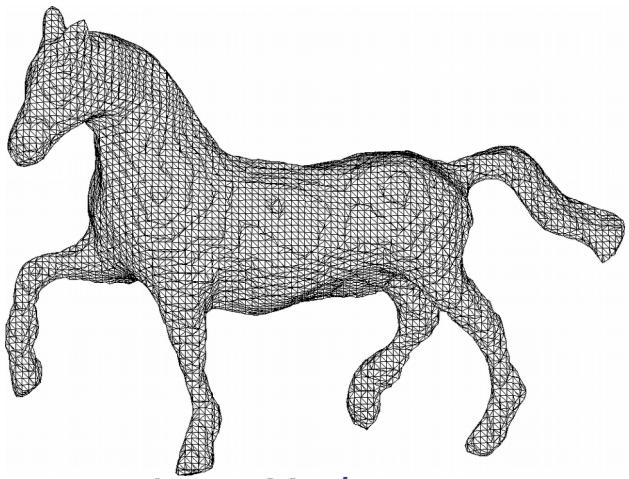
[X. Chen et al. SIGGRAPH 09]

Image segmentation and labeling

[Konishi and Yuille 00, Duygulu et al. 02, He et al. 04, Kumar and Hebert 03, Anguelov et al. 05, Tu et al.05, Schnitman et al. 06, Lim and Suter 07, Munoz et al. 08,...]

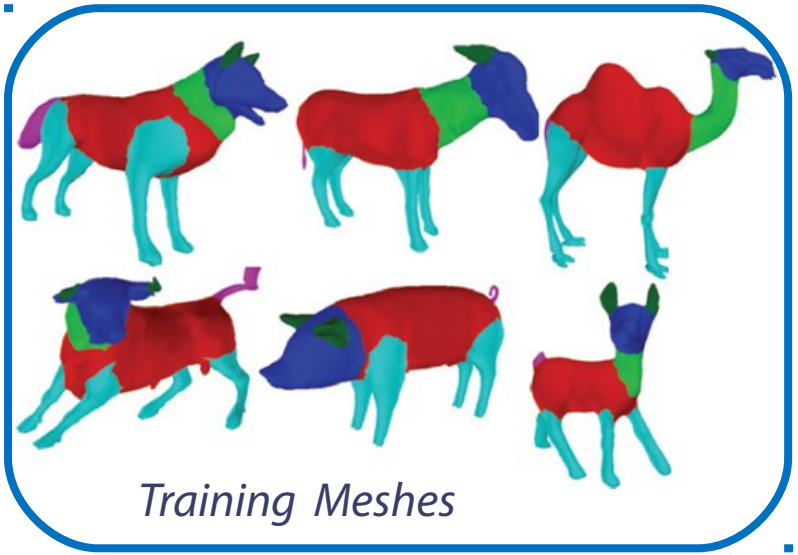
Textonboost
[Shotton et al. ECCV 06]

Shape Segmentation and Labeling



Input Mesh

Labeled Mesh



Training Meshes

Head
Neck
Torso
Leg
Tail
Ear

Kalogerakis, Hertzmann and Singh, 2010

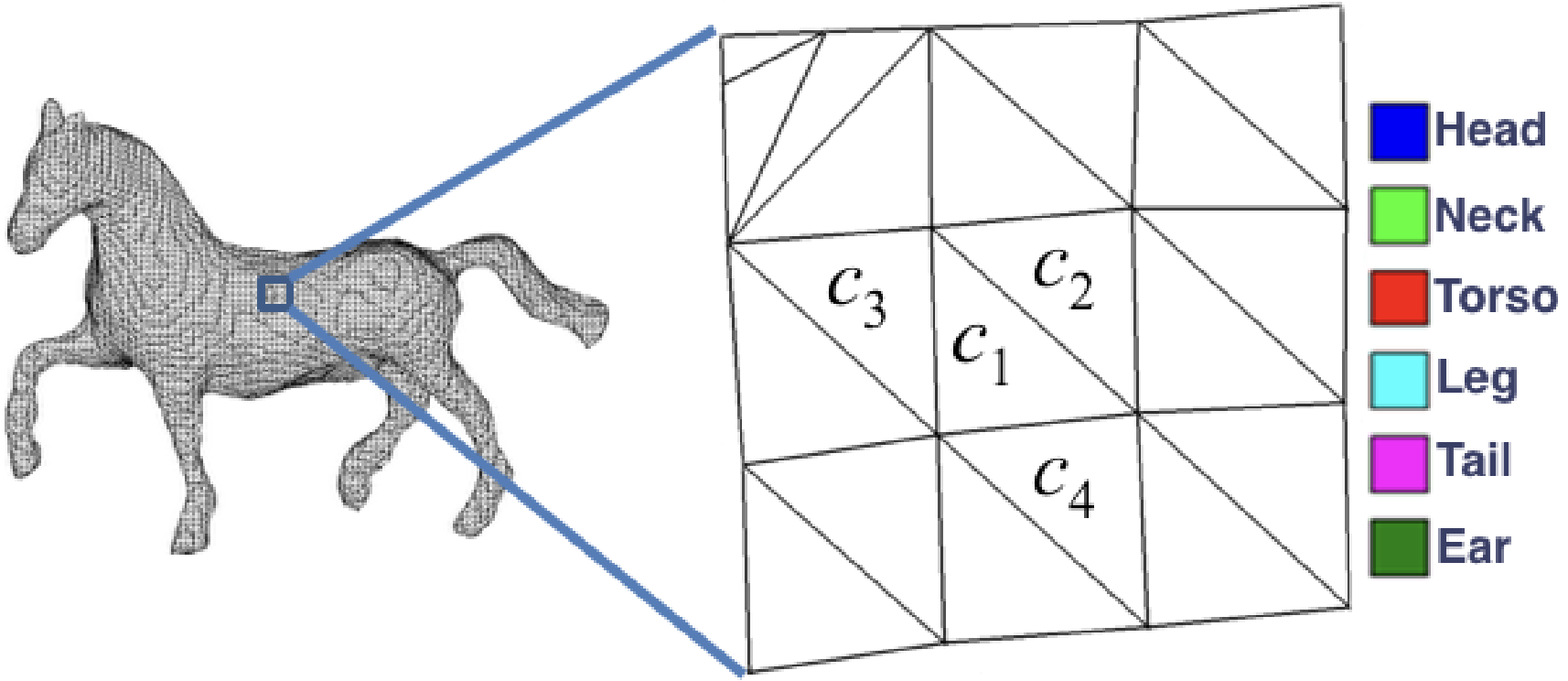
Why supervised segmentation?

- Learn from examples
- Significantly better results than state-of-the-art
- No manual parameter tuning
- Can learn different styles of segmentation

Kalogerakis, Hertzmann and Singh, 2010

Shape representation

- Polygon mesh, each polygonal face has label



$$c_1, c_2, c_3 \in Labels$$

$$Labels = \{\ head,\ neck,\ torso,\ leg,\ tail,\ ear\ \}$$

Key Insight #1

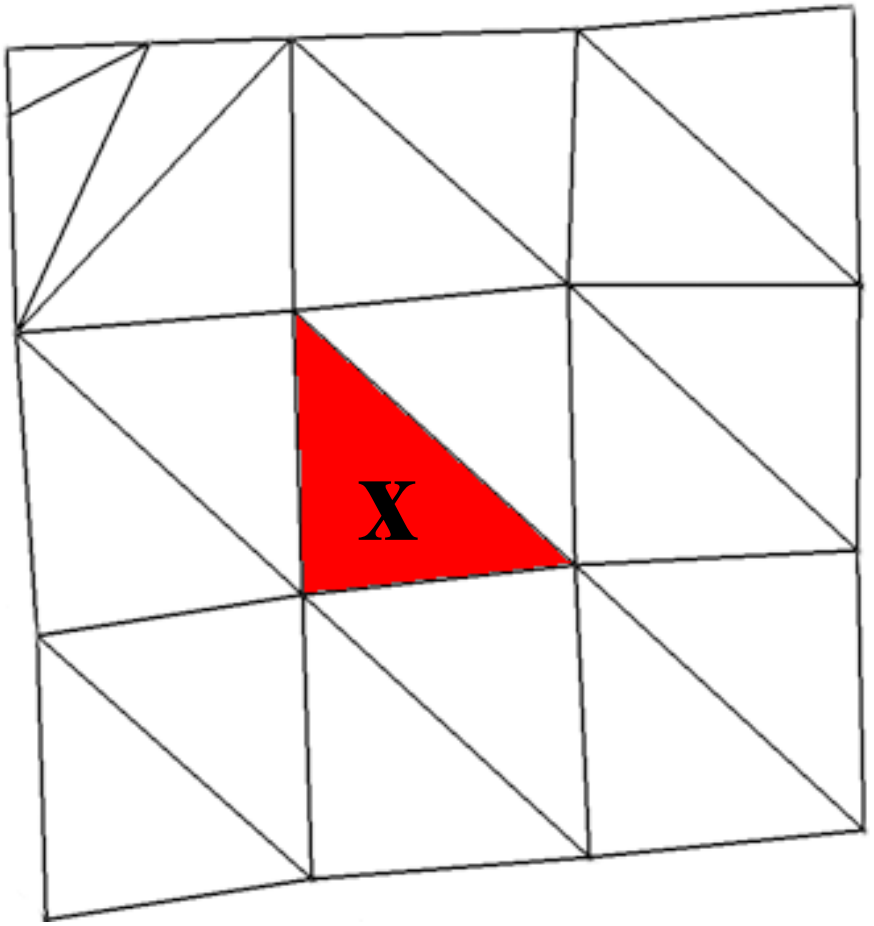
- Describe a face by **features** of its neighborhood
- Train a **classifier** to predict label from features

Kalogerakis, Hertzmann and Singh, 2010

Feature vector

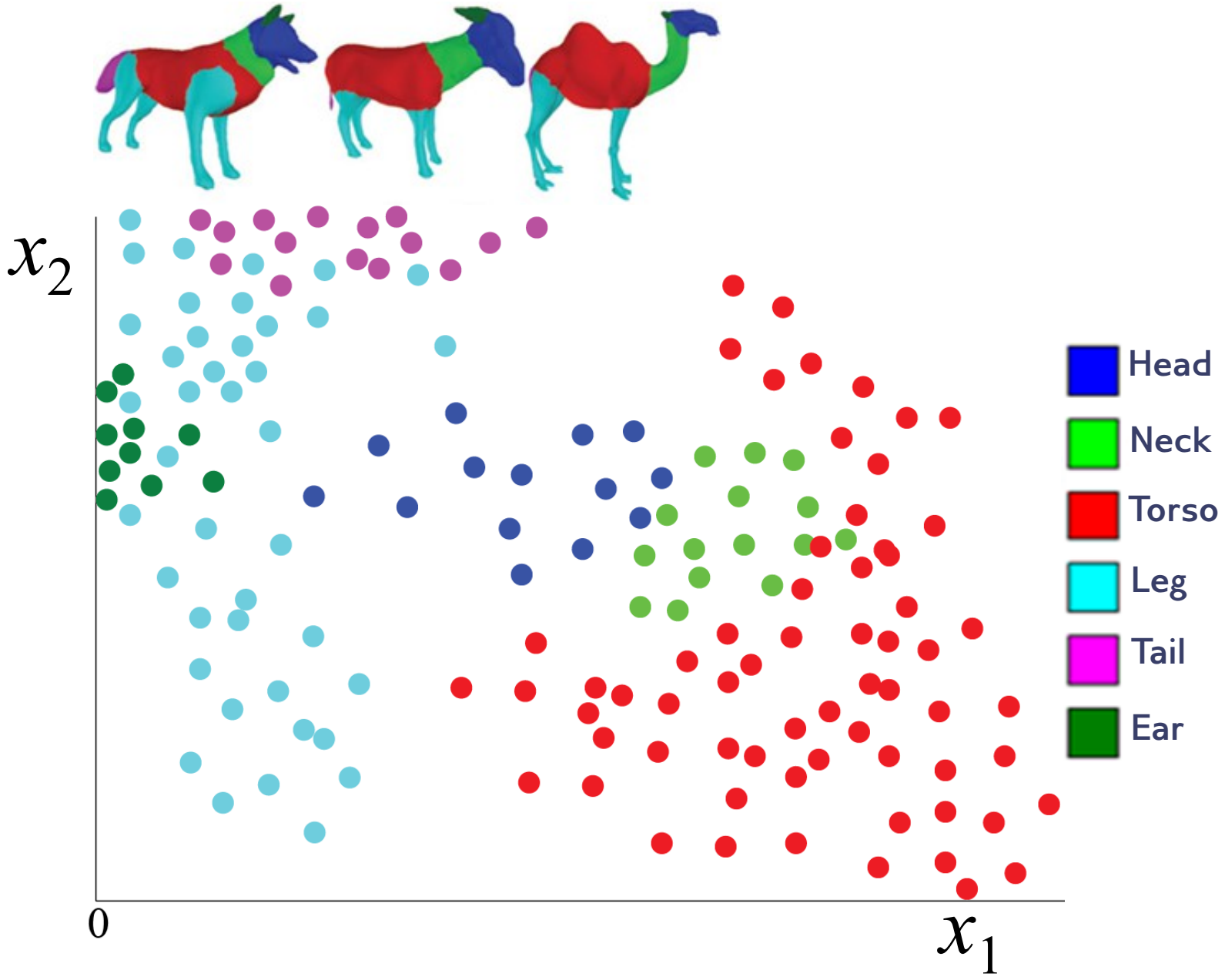
- surface curvature
- singular values from PCA
- shape diameter
- distances from medial surface
- average geodesic distances
- shape contexts
- spin images
- contextual label features

$$\mathbf{x} \in \Re^{375+35|C|} \rightarrow \mathrm{P}(c \mid \mathbf{x})$$



Kalogerakis, Hertzmann and Singh, 2010

Faces plotted in feature space

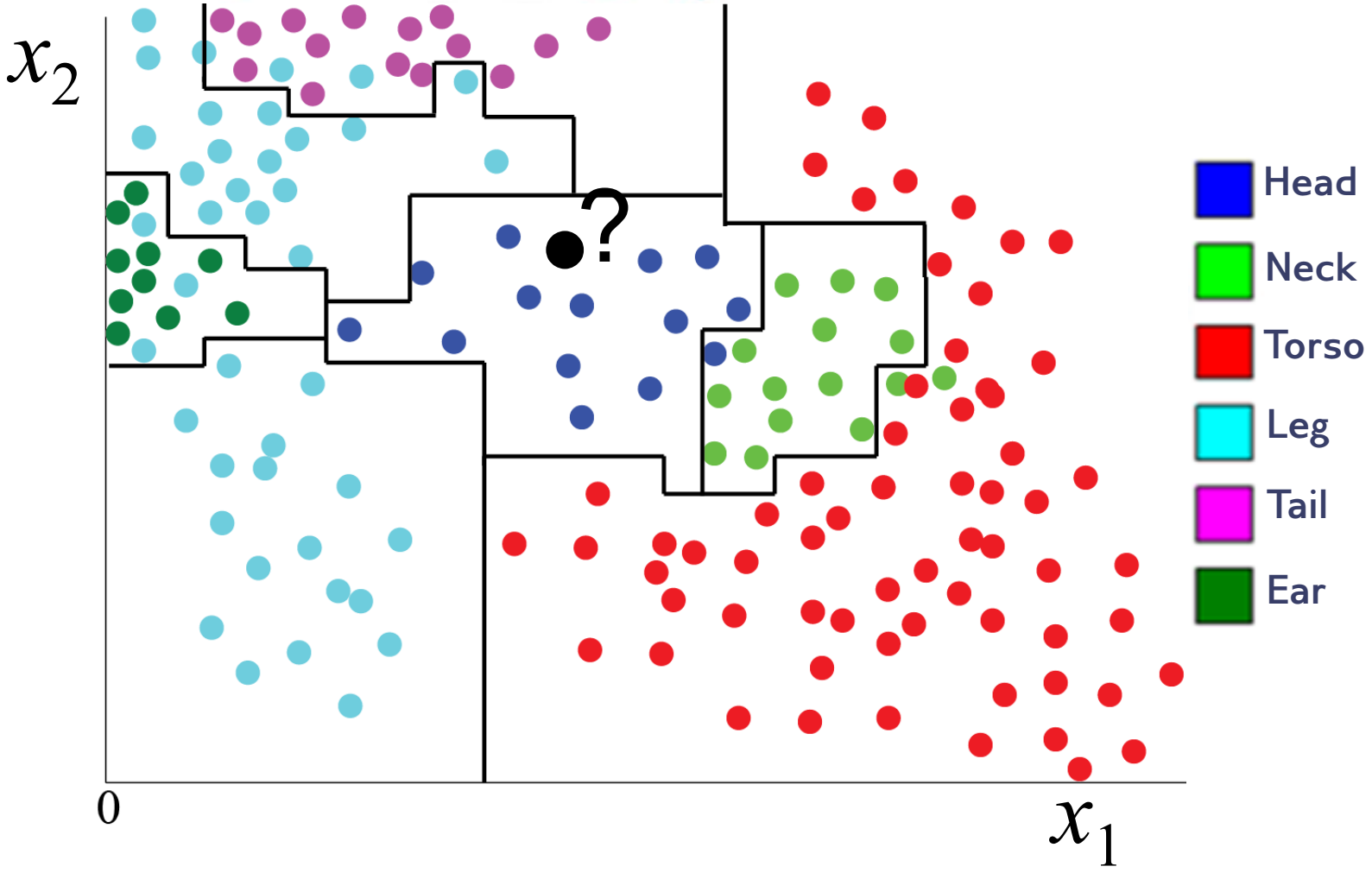


Kalogerakis, Hertzmann and Singh, 2010

Learning a classifier

JointBoost classifier

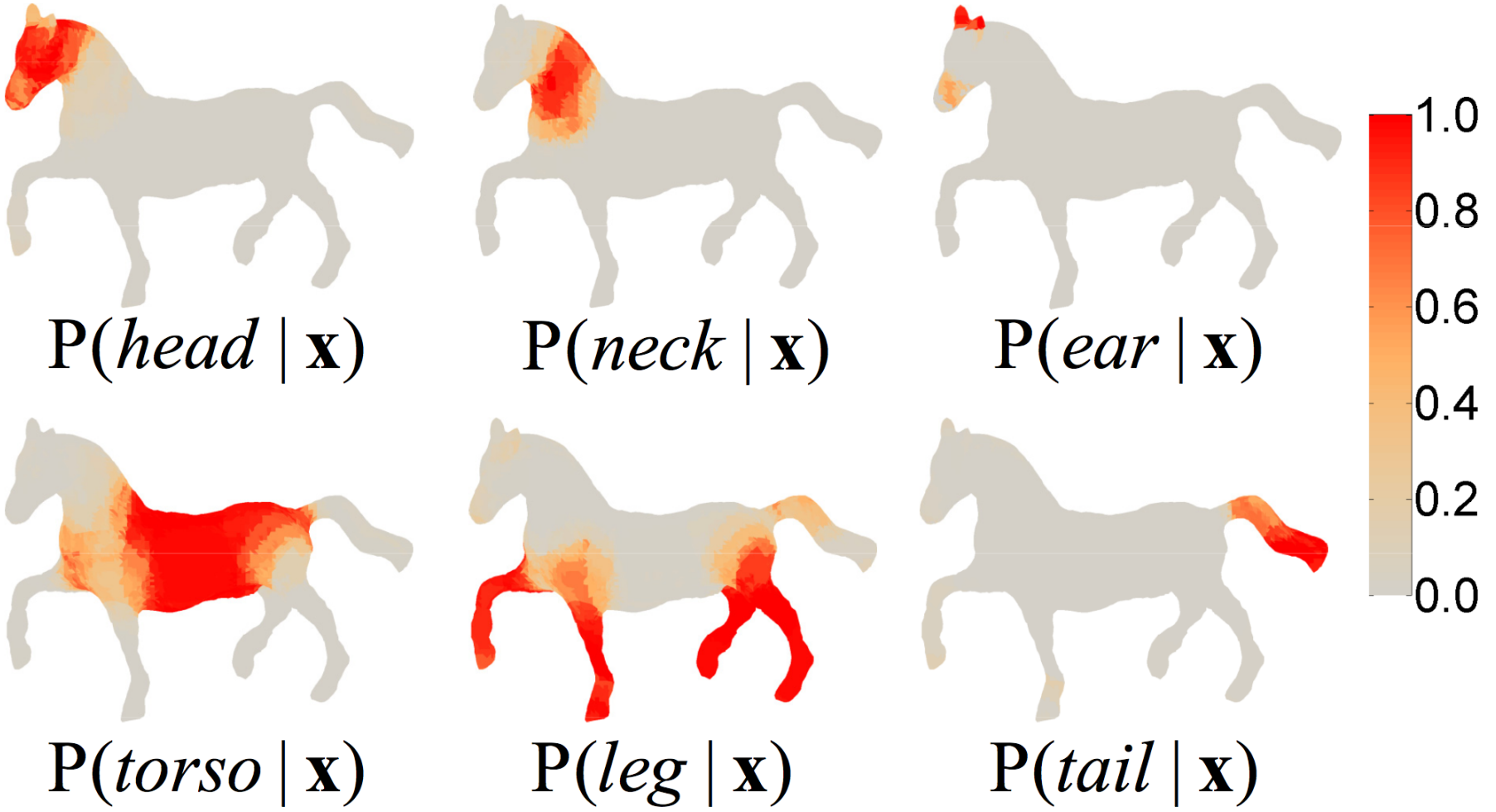
[Torralba et al. 2007]



Kalogerakis, Hertzmann and Singh, 2010

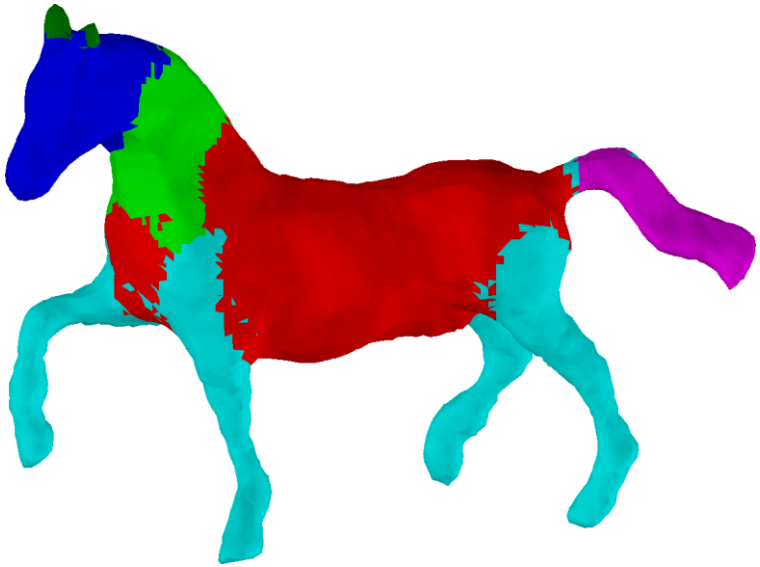
Classifier results

$$E_1(c;\mathbf{x}) = -\log P(c \mid \mathbf{x})$$

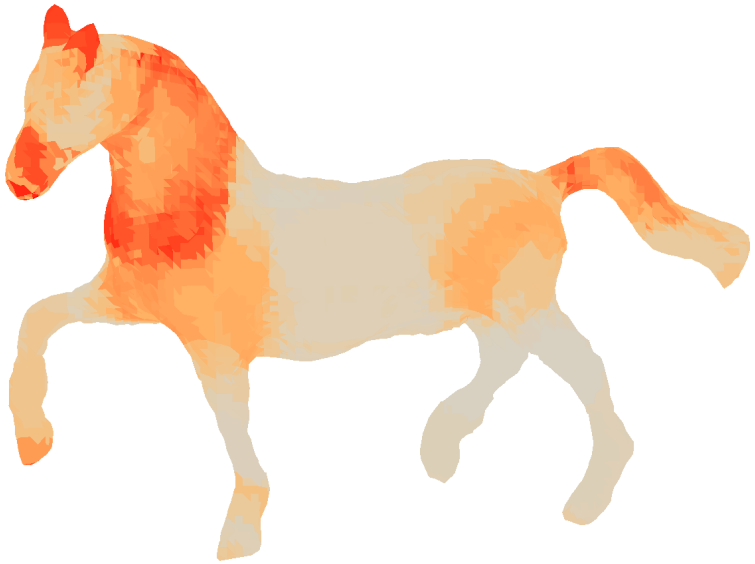


Kalogerakis, Hertzmann and Singh, 2010

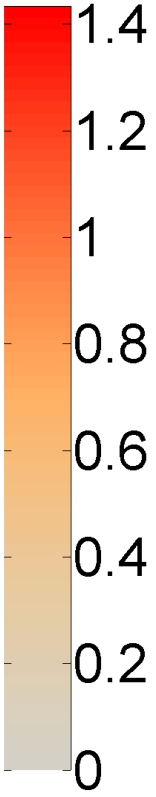
Classifier isn't great



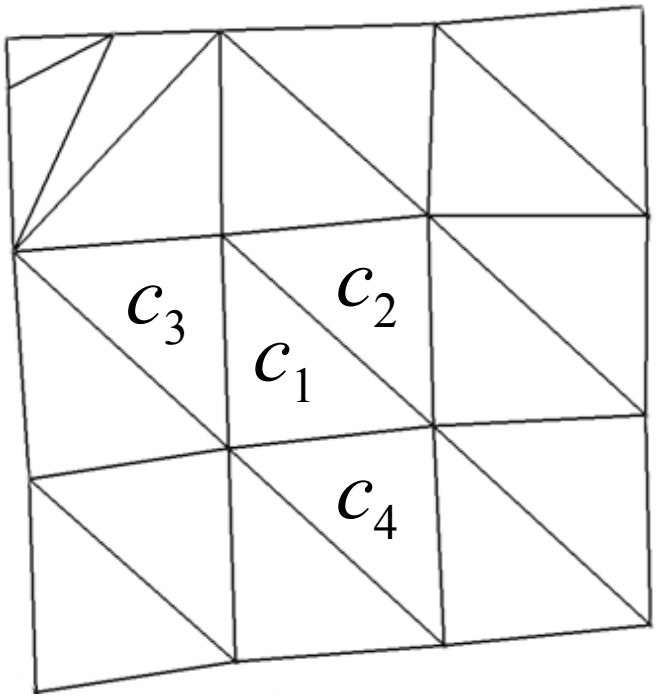
Most-likely labels



Classifier entropy



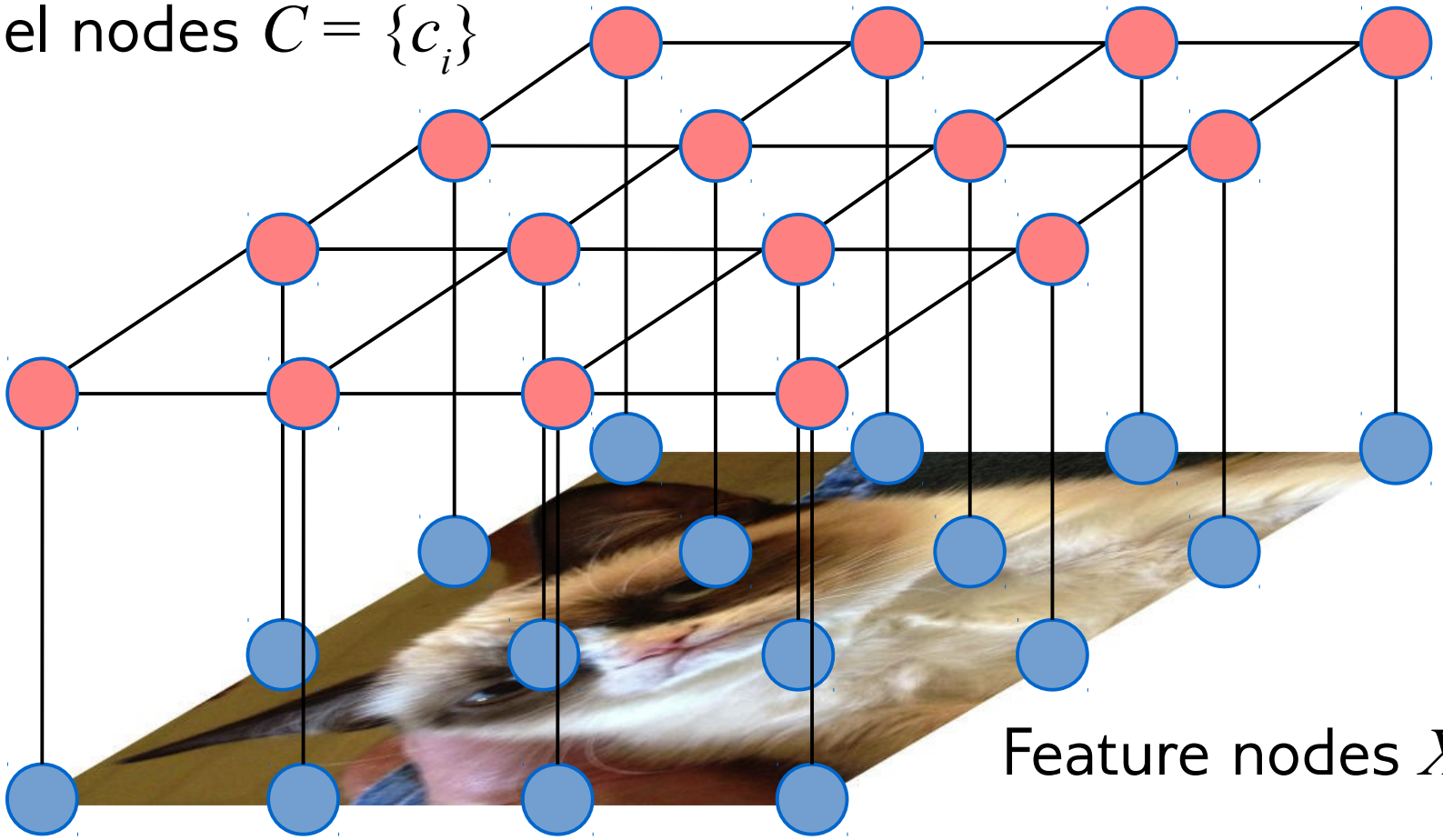
Key Insight #2



- Face's label is **correlated** with that of its neighbors
- **Similar** faces have similar labels

Conditional Random Field (CRF)

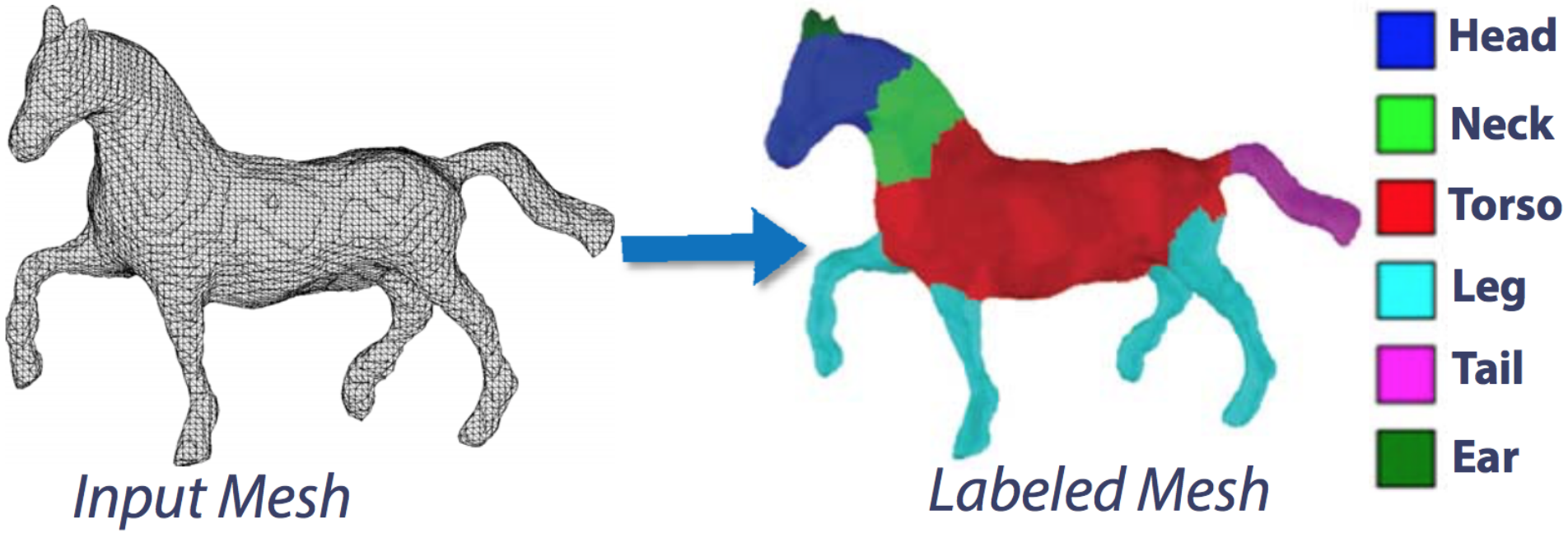
(an undirected probabilistic graphical model)



Factorized model of $\mathrm{P}(C \mid X) = \prod_i \mathrm{P}(c_i \mid X) = \prod_i \mathrm{P}(c_i \mid \text{nbrs of } c_i)$

Kalogerakis, Hertzmann and Singh, 2010

Conditional Random Field on Faces



$$c^* = \arg\min_{\mathbf{c}} \left\{ \sum_i \alpha_i \mathrm{E}_1(c_i; \mathbf{x_i}) + \sum_{i,j} l_{ij} \mathrm{E}_2(c_i, c_j; \mathbf{y_{ij}}) \right\}$$

Unary term

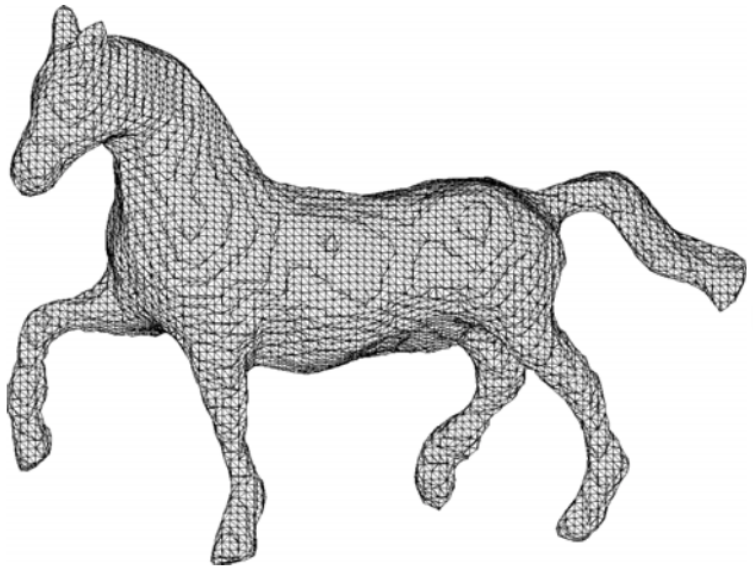
Pairwise term

Call this E for energy

Then $\mathrm{P}(C \mid X) = \exp(-E) / Z$, where Z is a normalization factor

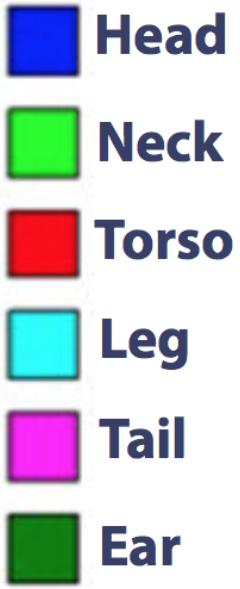
Kalogerakis, Hertzmann and Singh, 2010

Conditional Random Field on Faces



Input Mesh

Labeled Mesh

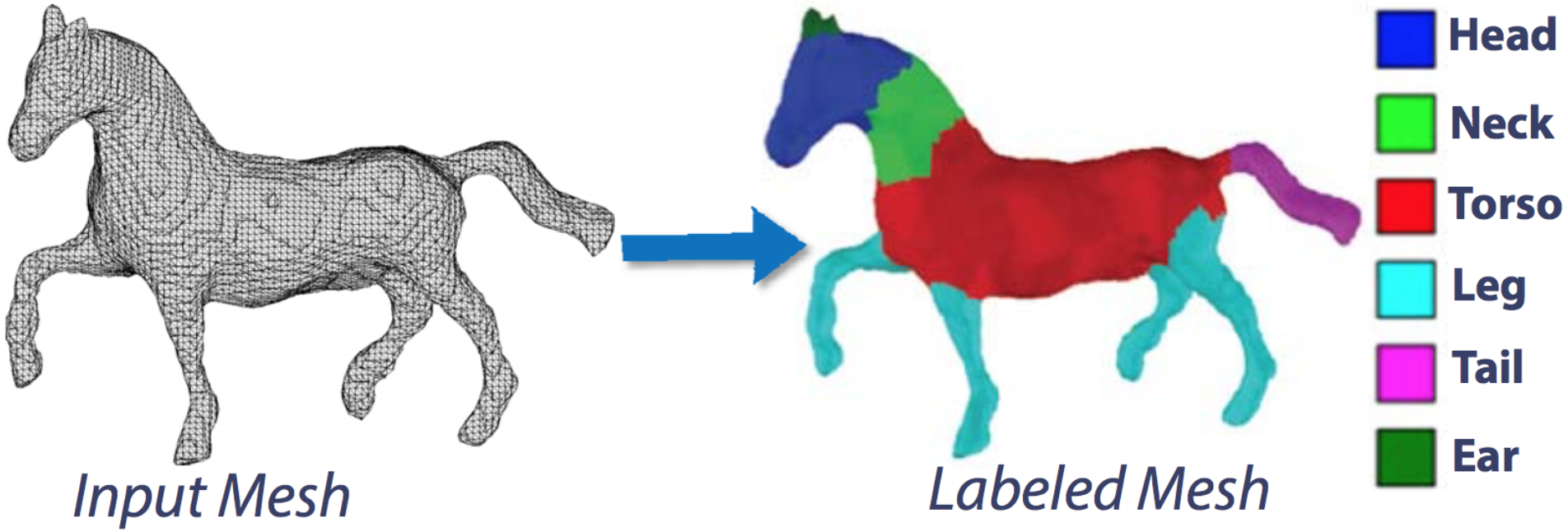


$$c^* = \arg\min_{\mathbf{c}} \left\{ \sum_i \alpha_i \mathrm{E}_1(c_i; \mathbf{x_i}) + \sum_{i,j} l_{ij} \mathrm{E}_2(c_i, c_j; \mathbf{y_{ij}}) \right\}$$

Unary term

Kalogerakis, Hertzmann and Singh, 2010

Conditional Random Field on Faces

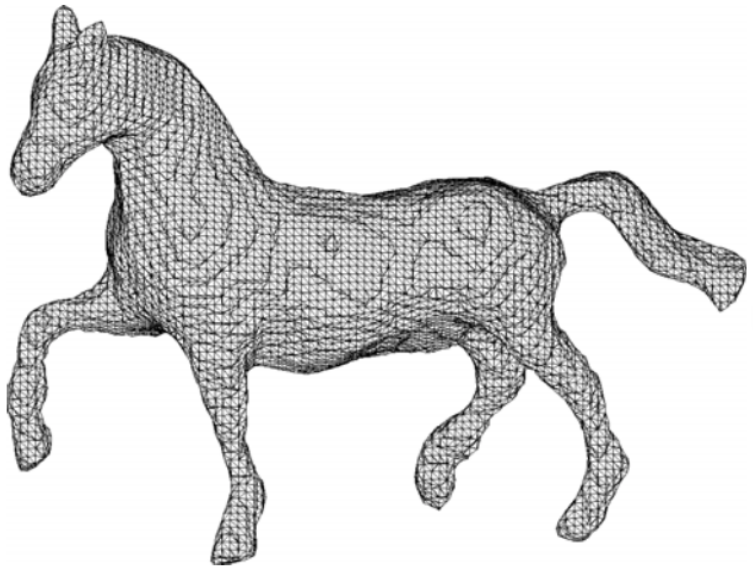


$$c^* = \arg\min_{\mathbf{c}} \left\{ \sum_i \alpha_i \mathrm{E}_1(c_i; \mathbf{x_i}) + \sum_{i,j} l_{ij} \mathrm{E}_2(c_i, c_j; \mathbf{y_{ij}}) \right\}$$

Face features

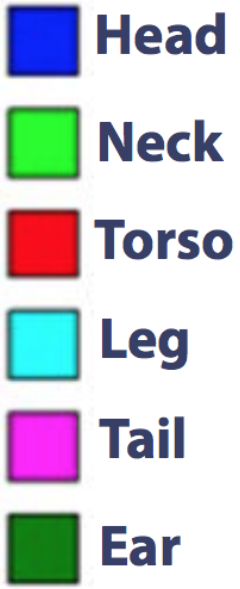
Kalogerakis, Hertzmann and Singh, 2010

Conditional Random Field on Faces



Input Mesh

Labeled Mesh

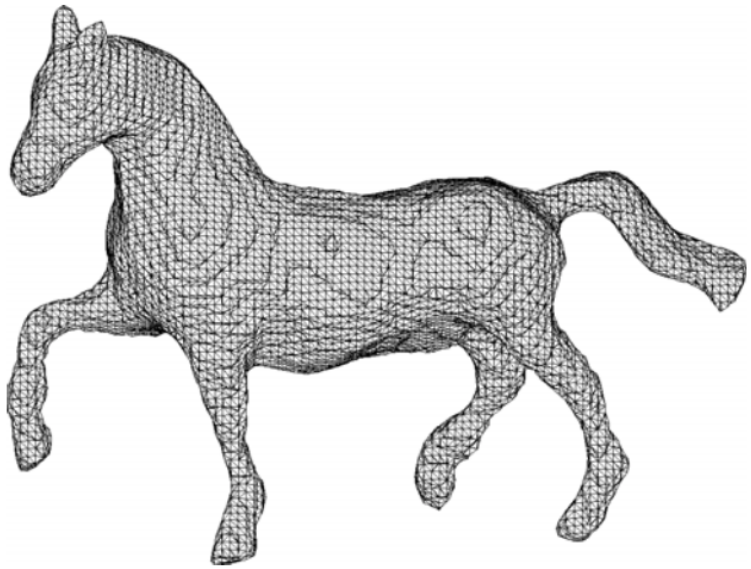


$$c^* = \arg\min_{\mathbf{c}} \left\{ \sum_i \alpha_i \mathrm{E}_1(c_i; \mathbf{x_i}) + \sum_{i,j} l_{ij} \mathrm{E}_2(c_i, c_j; \mathbf{y_{ij}}) \right\}$$

Face area

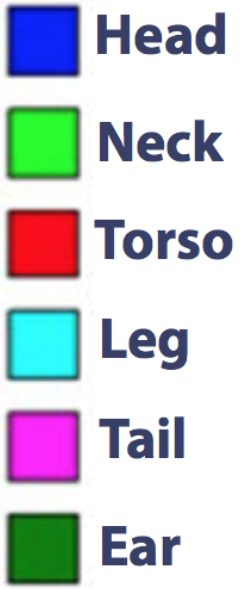
Kalogerakis, Hertzmann and Singh, 2010

Conditional Random Field on Faces



Input Mesh

Labeled Mesh

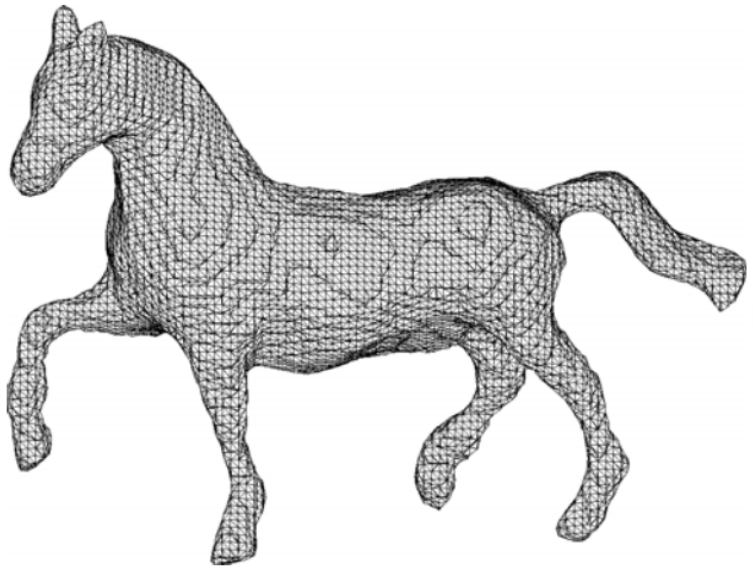


$$c^* = \arg\min_{\mathbf{c}} \left\{ \sum_i \alpha_i \mathrm{E}_1(c_i; \mathbf{x_i}) + \sum_{i,j} l_{ij} \mathrm{E}_2(c_i, c_j; \mathbf{y_{ij}}) \right\}$$

Pairwise term

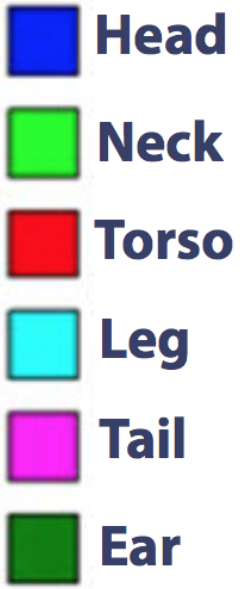
Kalogerakis, Hertzmann and Singh, 2010

Conditional Random Field on Faces



Input Mesh

Labeled Mesh

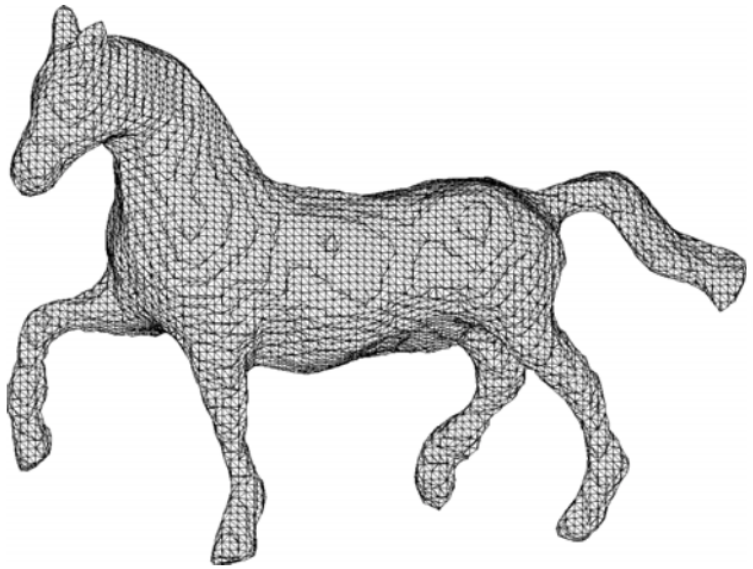


$$c^* = \arg\min_{\mathbf{c}} \left\{ \sum_{i} \alpha_i \mathrm{E}_1(c_i; \mathbf{x_i}) + \sum_{i,j} l_{ij} \mathrm{E}_2(c_i, c_j; \mathbf{y_{ij}}) \right\}$$

Edge features

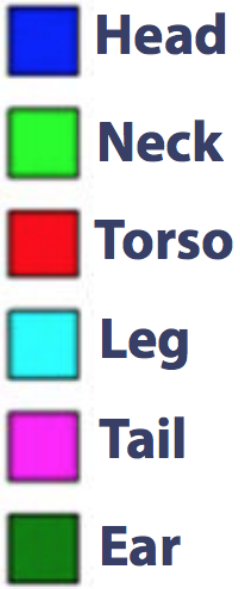
Kalogerakis, Hertzmann and Singh, 2010

Conditional Random Field on Faces



Input Mesh

Labeled Mesh



$$c^* = \arg\min_{\mathbf{c}} \left\{ \sum_i \alpha_i \mathrm{E}_1(c_i; \mathbf{x_i}) + \sum_{i,j} l_{ij} \mathrm{E}_2(c_i, c_j; \mathbf{y_{ij}}) \right\}$$

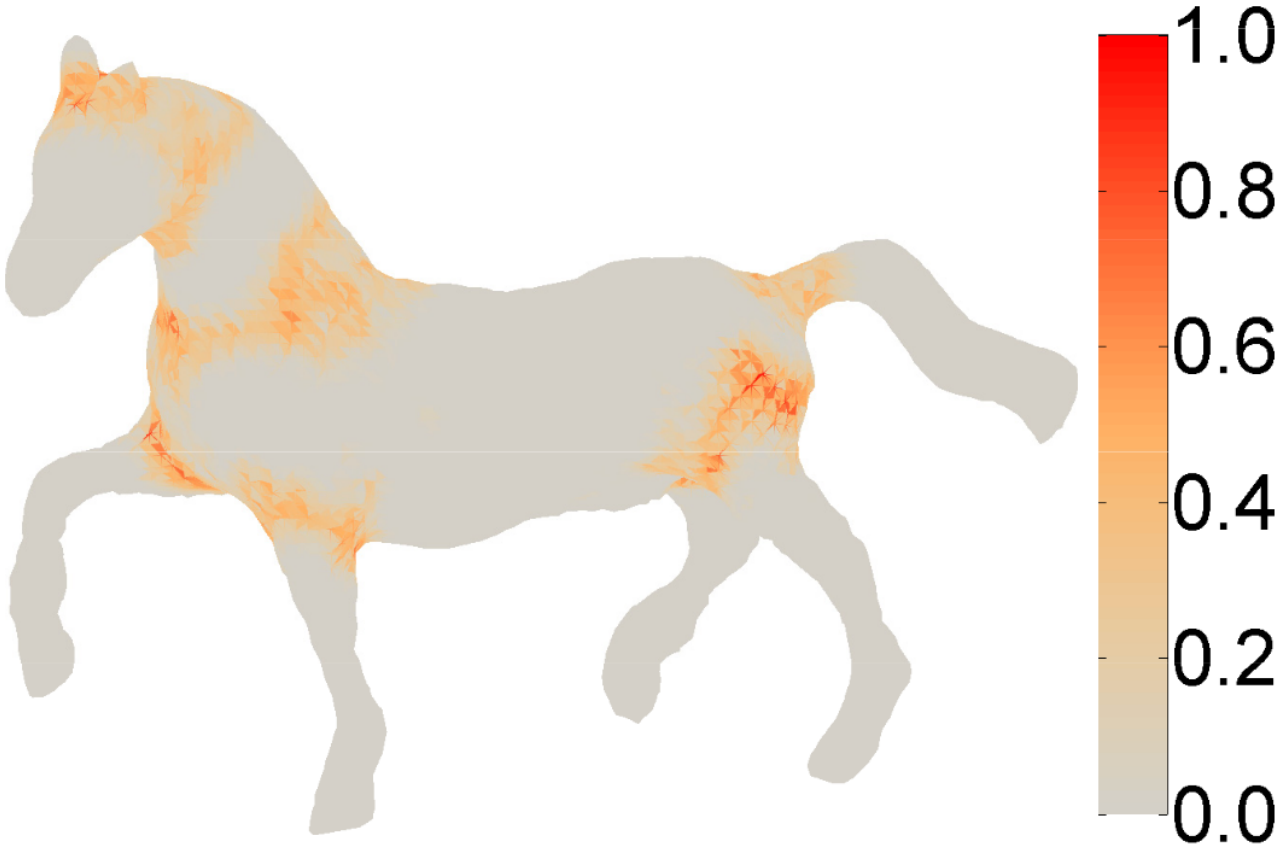
Edge length

Kalogerakis, Hertzmann and Singh, 2010

Pairwise Term

$$\mathrm{E}_2(c,c';\mathbf{y},\theta_2) = G(\mathbf{y})L(c,c')$$

Geometry-dependent term



Kalogerakis, Hertzmann and Singh, 2010

Pairwise Term

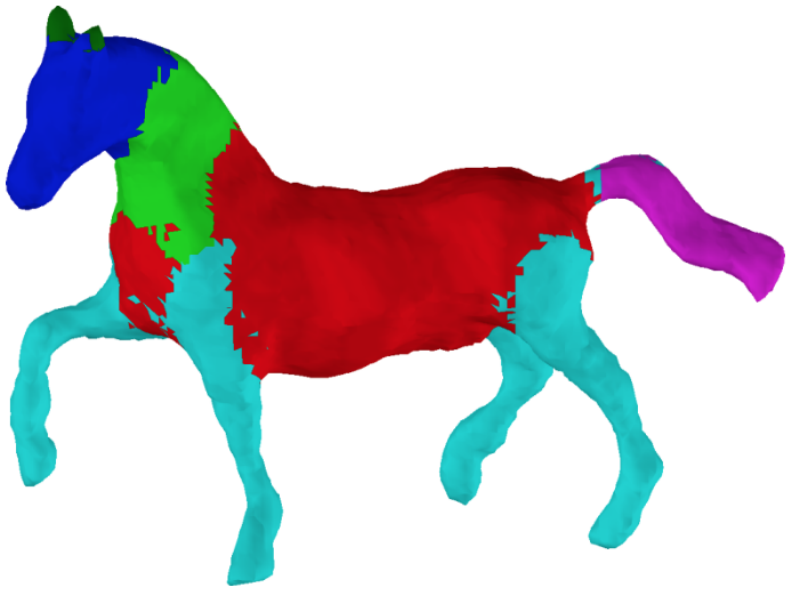
$$\mathrm{E}_2(c,c';\mathbf{y},\theta_2) = G(\mathbf{y})L(c,c')$$

Label compatibility term

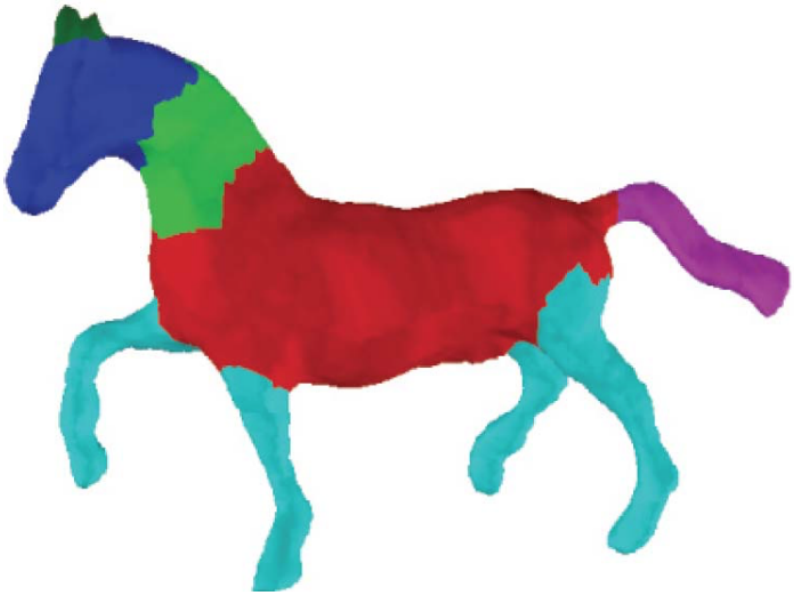
$$L(c,c') = \begin{array}{c} \\ \text{Head} \\ \text{Neck} \\ \text{Ear} \\ \text{Torso} \\ \text{Leg} \\ \text{Tail} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{Head} & \text{Neck} & \text{Ear} & \text{Torso} & \text{Leg} & \text{Tail} \end{array} \\ \begin{bmatrix} 0 & .45 & .07 & 1 & \infty & \infty \\ .45 & 0 & \infty & 1 & \infty & \infty \\ .07 & \infty & 0 & \infty & \infty & \infty \\ 1 & 1 & \infty & 0 & 1 & .56 \\ \infty & \infty & \infty & 1 & 0 & \infty \\ \infty & \infty & \infty & .56 & \infty & 0 \end{bmatrix} \end{array}$$

Kalogerakis, Hertzmann and Singh, 2010

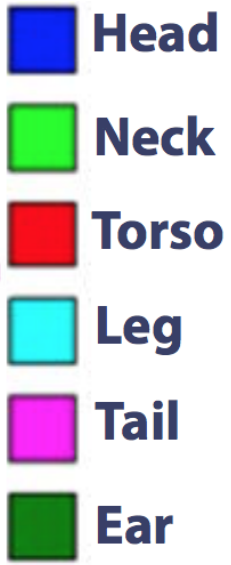
Effect of the pairwise term



Unary term classifier

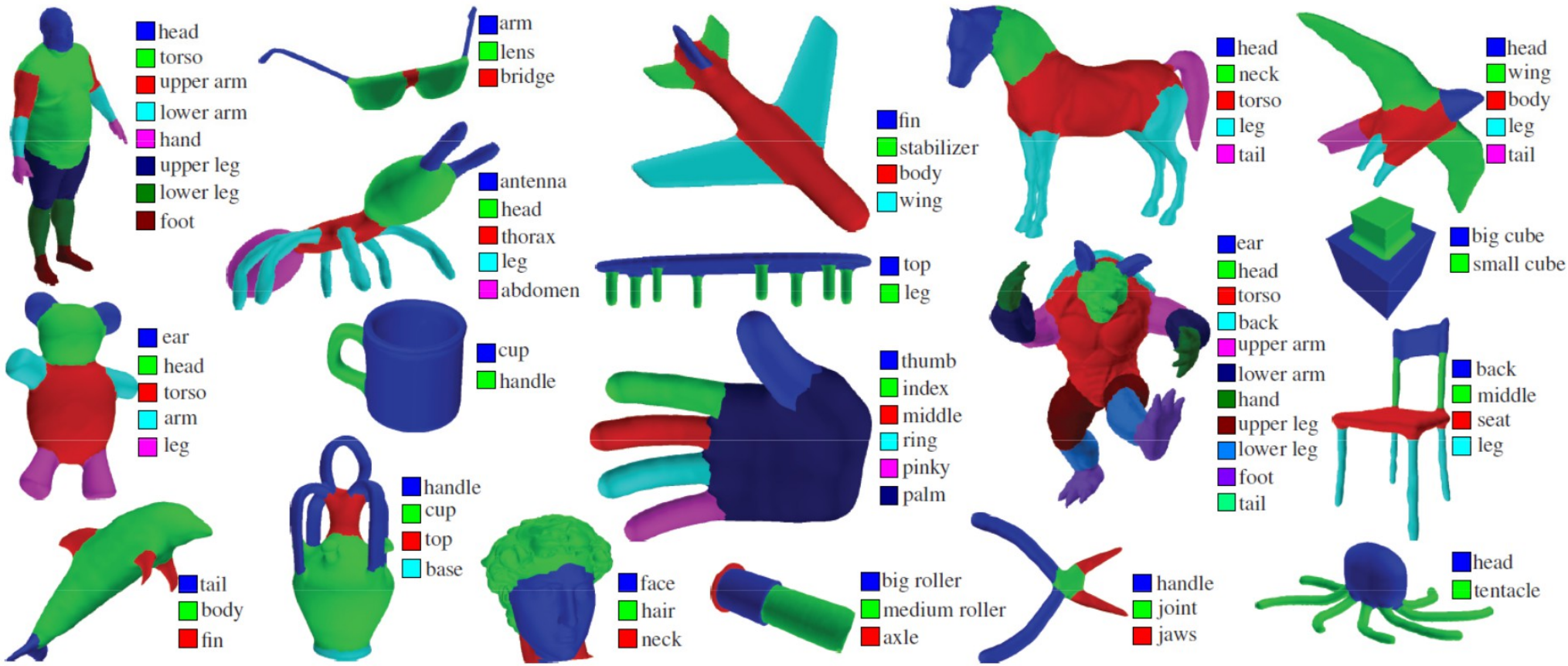


Full CRF result



Kalogerakis, Hertzmann and Singh, 2010

More results



94% labeling accuracy, 9.5% segmentation error (prev best 16%)

Training and Inference

- Training:
  - (Joint-) Boosting for unary classifier
  - Holdout validation with grid search and gradient descent for remaining parameters

- Inference (finding the optimal face labeling by minimizing the CRF energy):
  - α-expansion graph cuts

Key Insight #3

- A **better unary classifier** would improve results
- The best current visual classifiers, in general, are **convolutional neural networks (CNNs)**

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

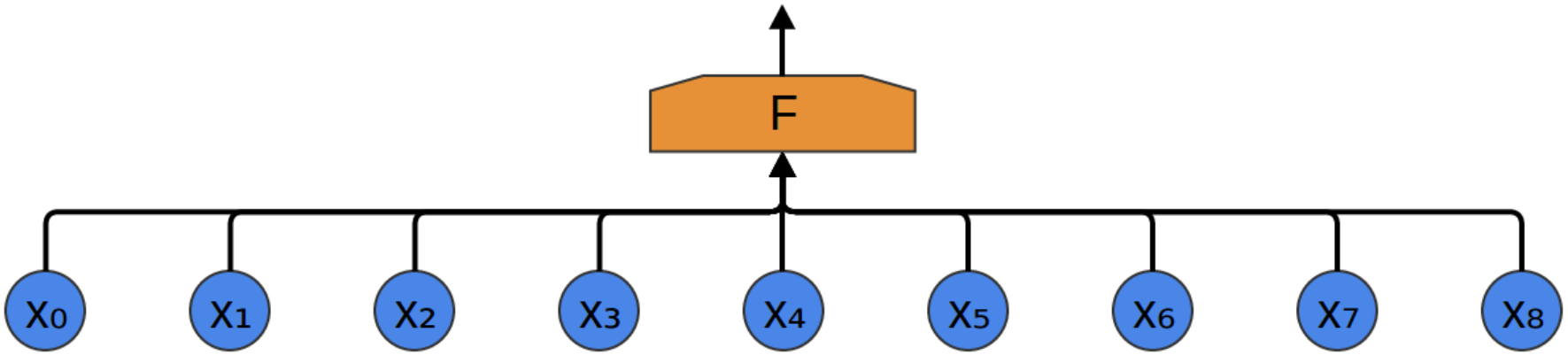
In 2012, the error rate in the ImageNet visual recognition challenge was halved by a deep CNN (gains are typically incremental). There are 1000 categories: the baseline of random guessing would have a 99.9% error.

What is a Convolutional Neural Network?

- Imagine we have a set of N samples from some signal
- We want to produce a prediction, e.g. whether the signal represents a human voice, or a picture of a cat, or a depth image of a building

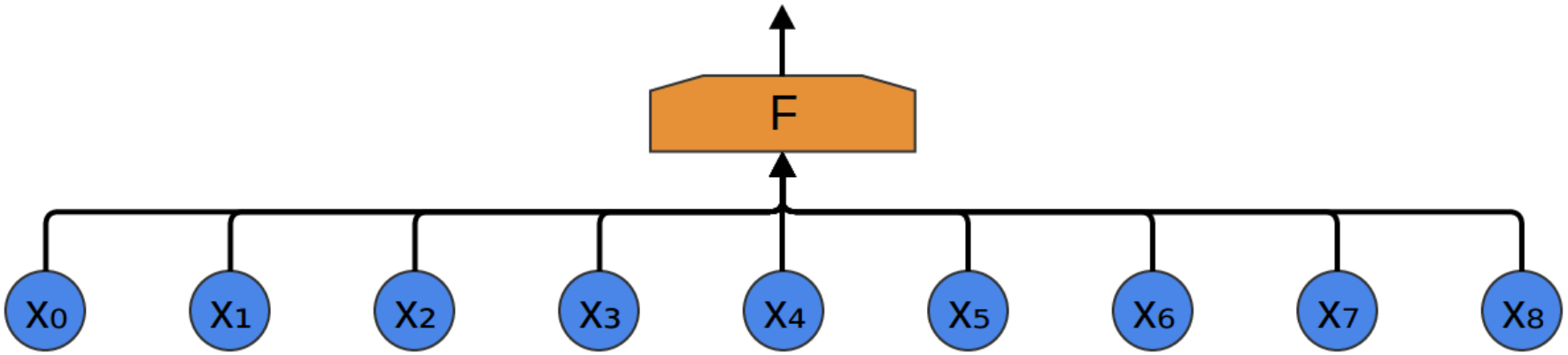
What is a Convolutional Neural Network?

- We can compute the probability as a function F of these values
  - In a **fully-connected** network, the function takes in all the inputs at once, e.g. as $g(\mathbf{w}\cdot\mathbf{x})$, where $\mathbf{w}$ is a weight vector and g is some nonlinear transformation such as a sigmoid function



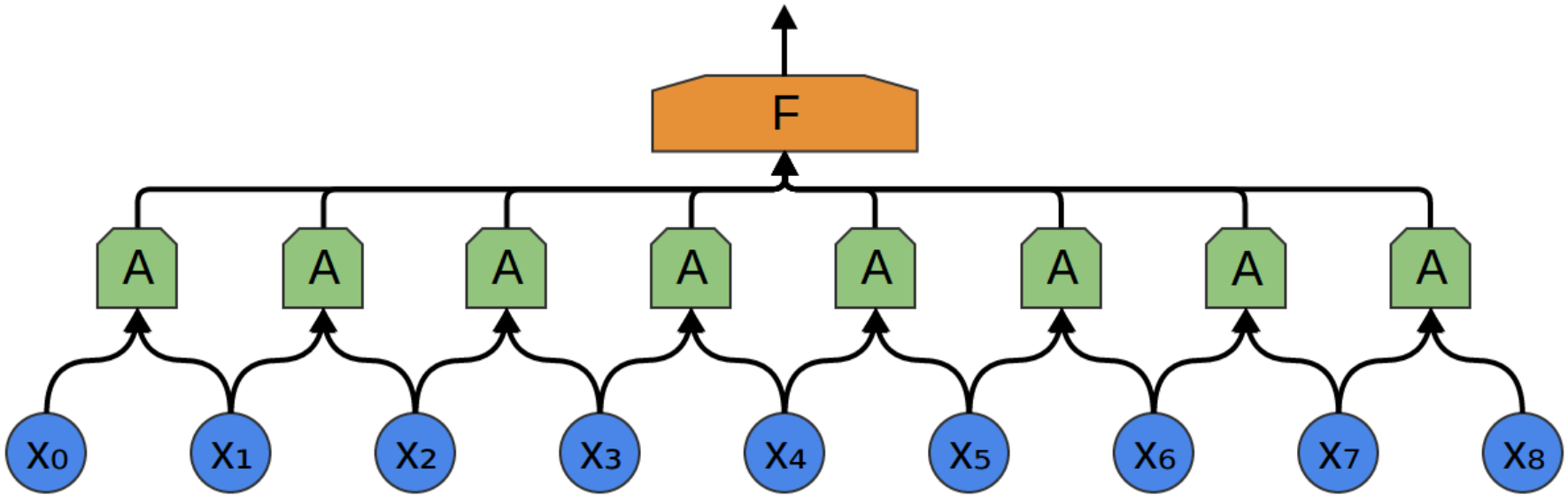
What is a Convolutional Neural Network?

- Fully-connected networks have some drawbacks
  - The function is **very high-dimensional** (all inputs processed at once)
  - **No complex relationships** between inputs are modeled (just a dot product)
  - Local information is **not captured in a "translation-invariant" way** (a feature of the signal at the left end of the sequence must be learned independently of the same feature occurring at the right end)



What is a Convolutional Neural Network?

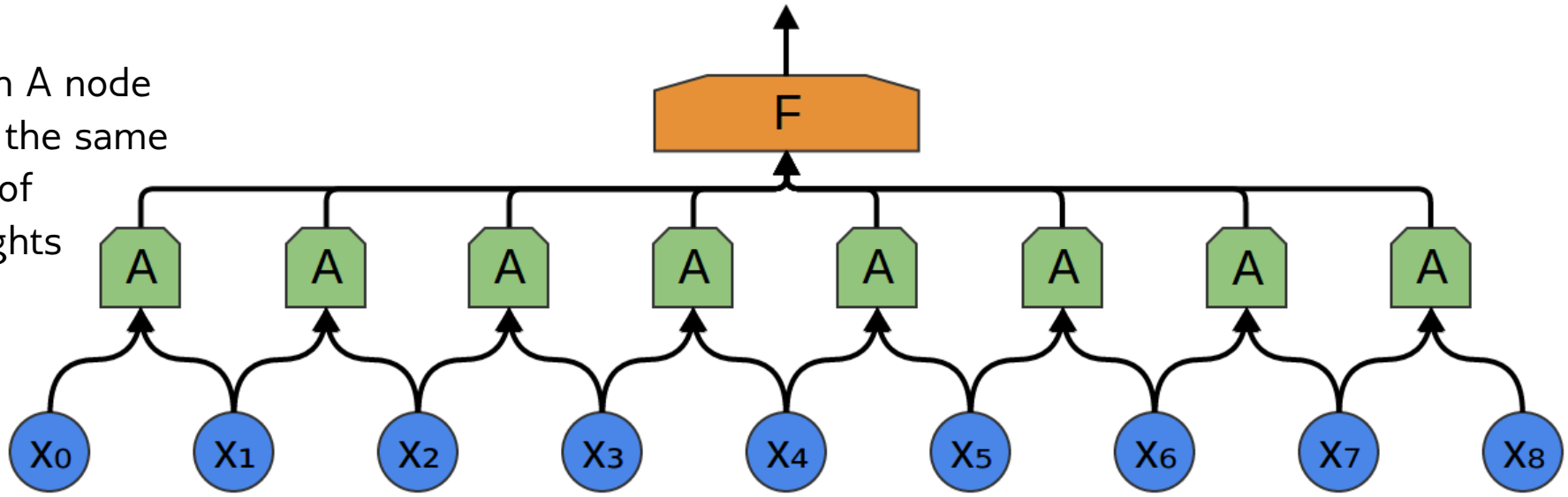
- **Solution:** a **convolutional layer**
- A filter (again, a dot product followed by a nonlinear transformation) is applied on local neighborhoods of the signal



What is a Convolutional Neural Network?

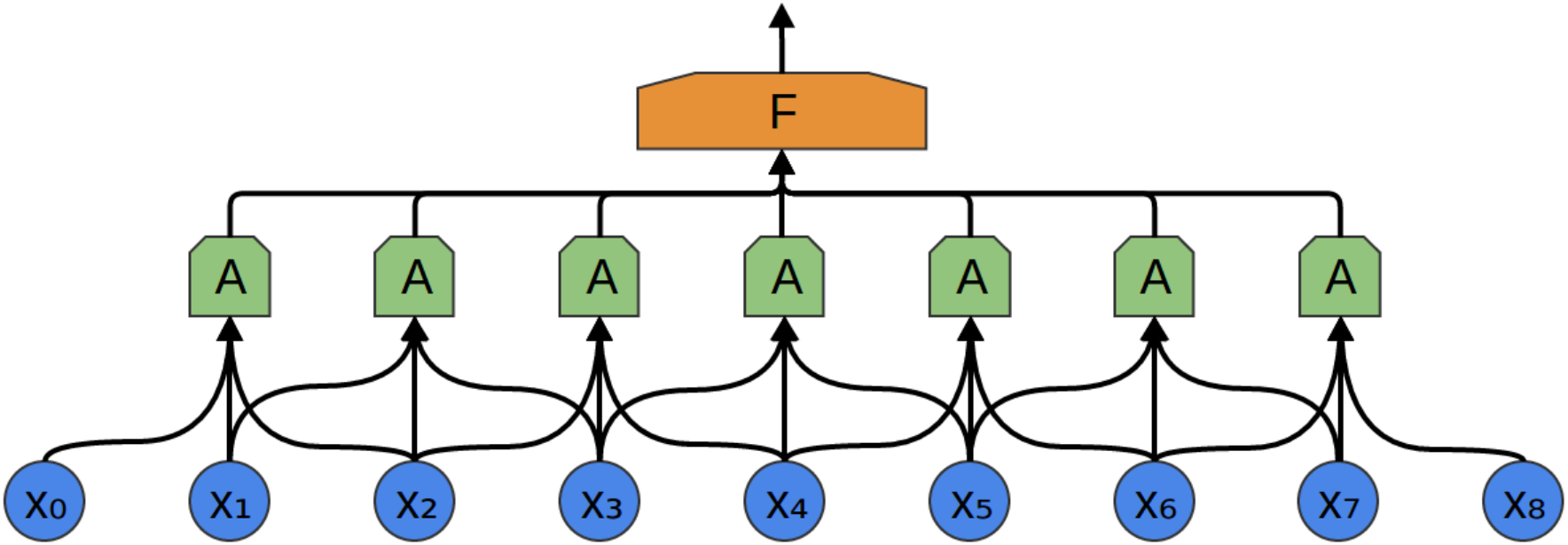
- All filters **share the same weights**!
  - Dramatically reduces number of parameters of the network
- The final output is a function of the filter responses

Each A node has the same set of weights



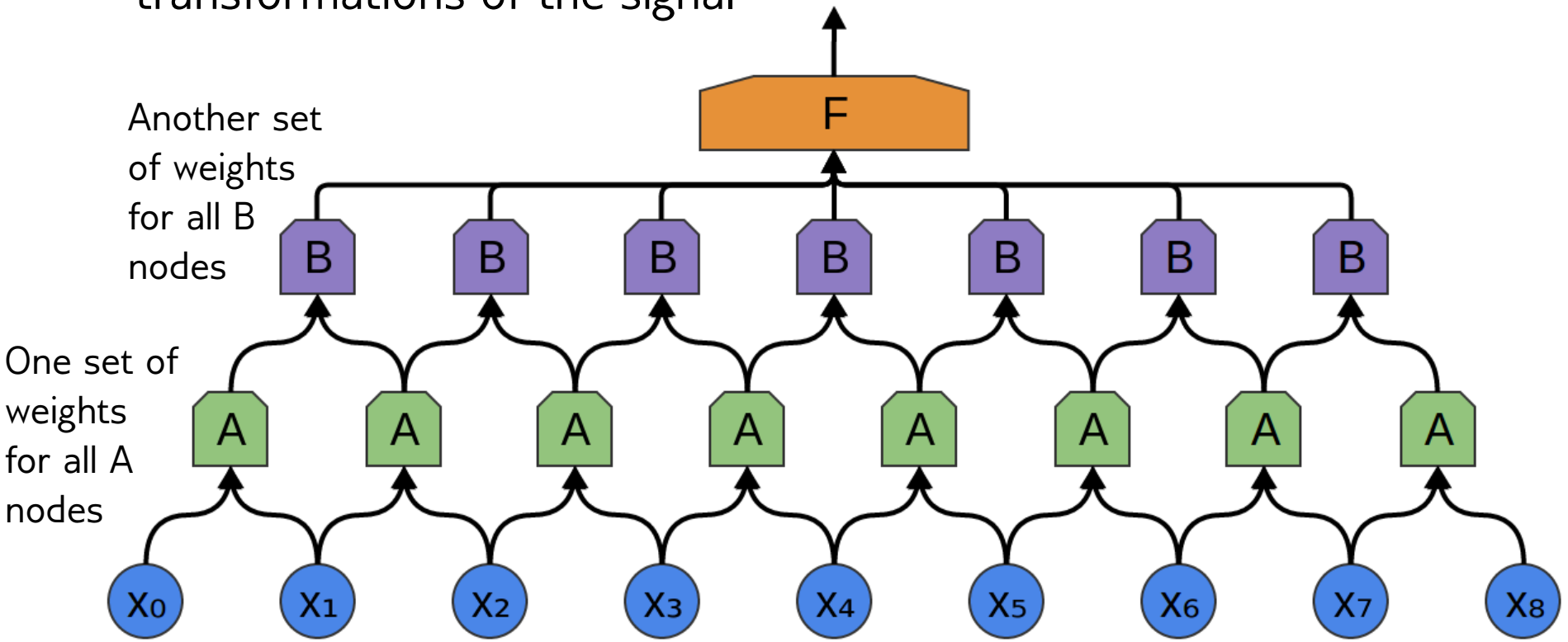
What is a Convolutional Neural Network?

- We can make the neighborhoods larger, to capture broader local features



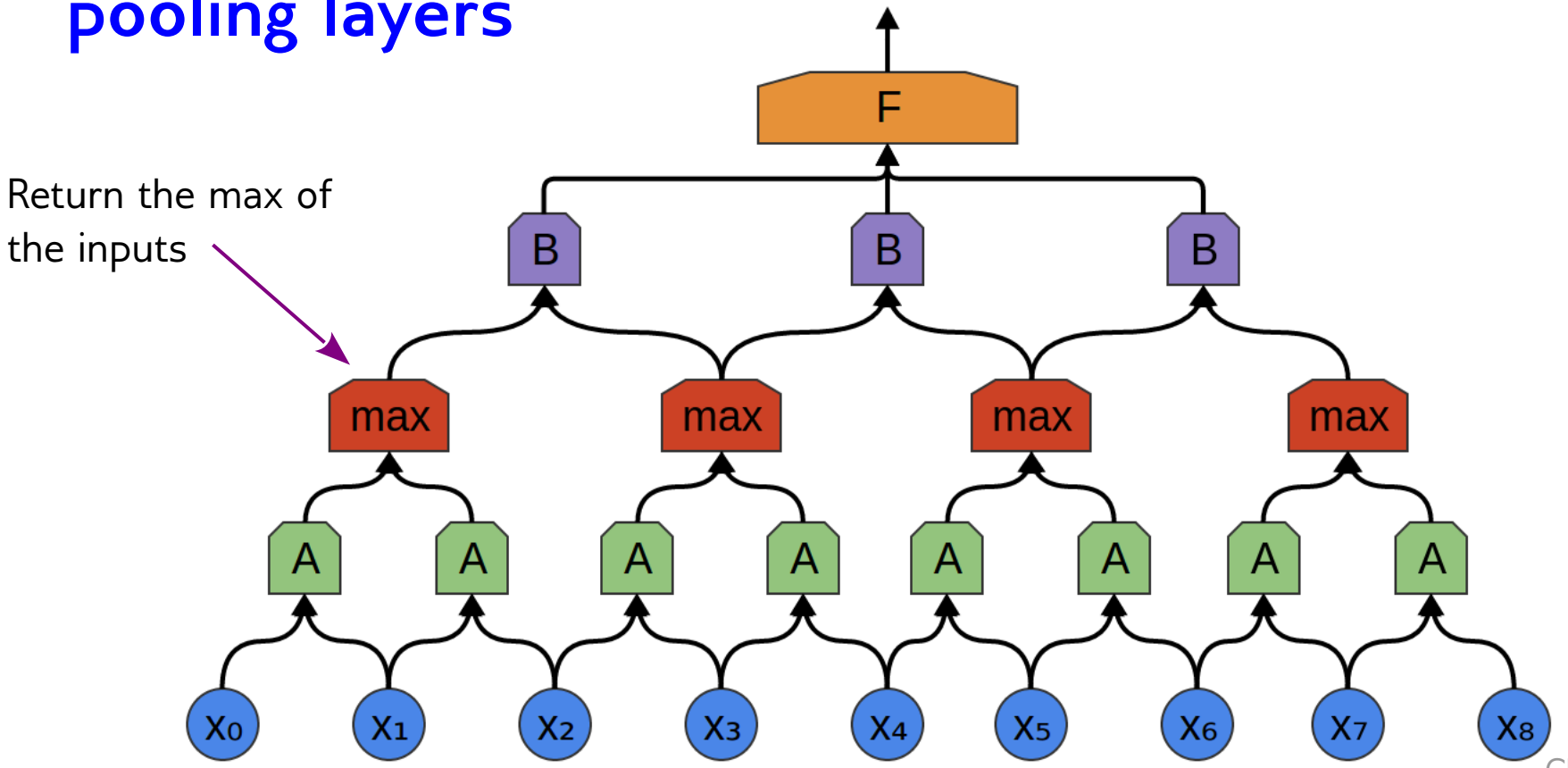
What is a Convolutional Neural Network?

- Convolutional layers are **composable**: they can be stacked with each layer providing inputs for the next layer
  - Higher layers can capture more abstract features since they effectively cover larger neighborhoods, and combine multiple different nonlinear transformations of the signal



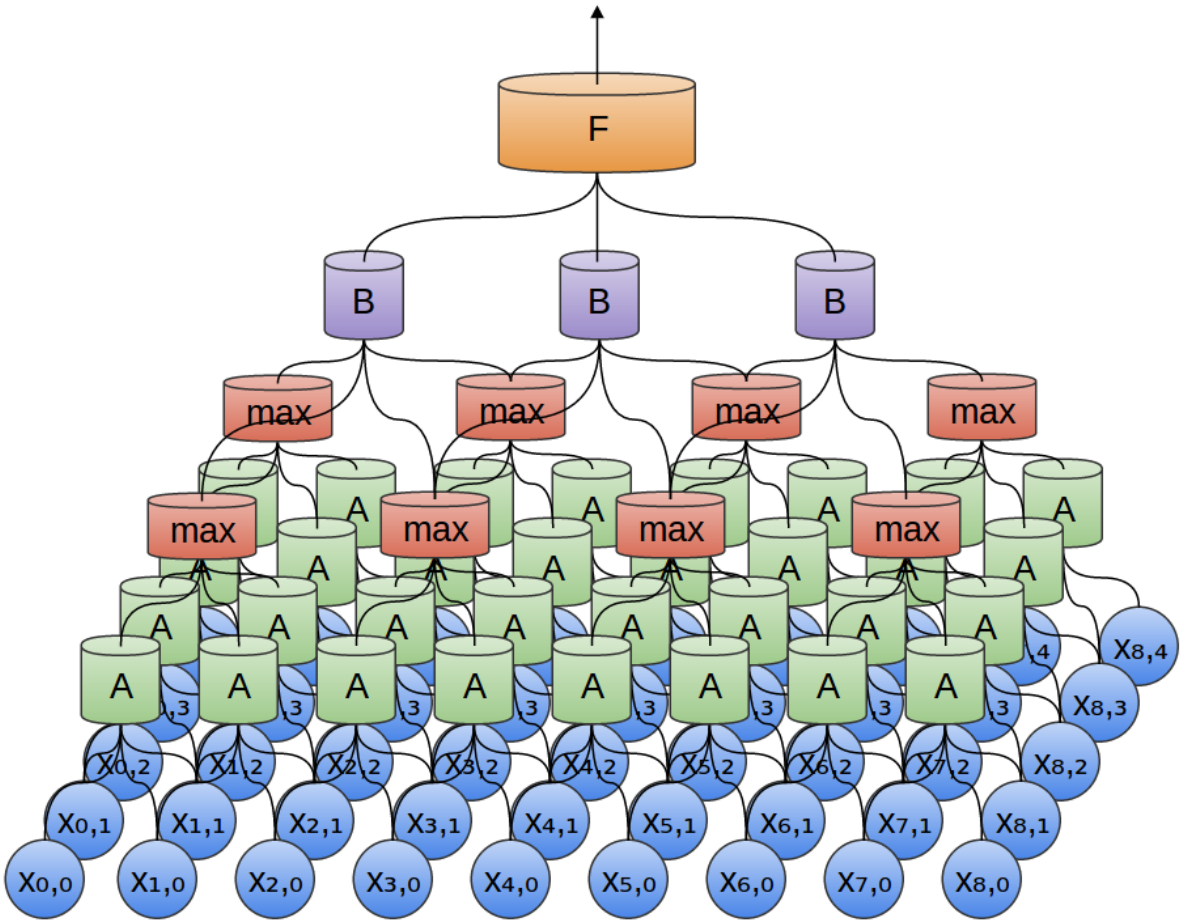
What is a Convolutional Neural Network?

- To make the network robust to small translations in detected features, and to reduce the amount of redundant data fed into higher layers, we introduce **pooling layers**



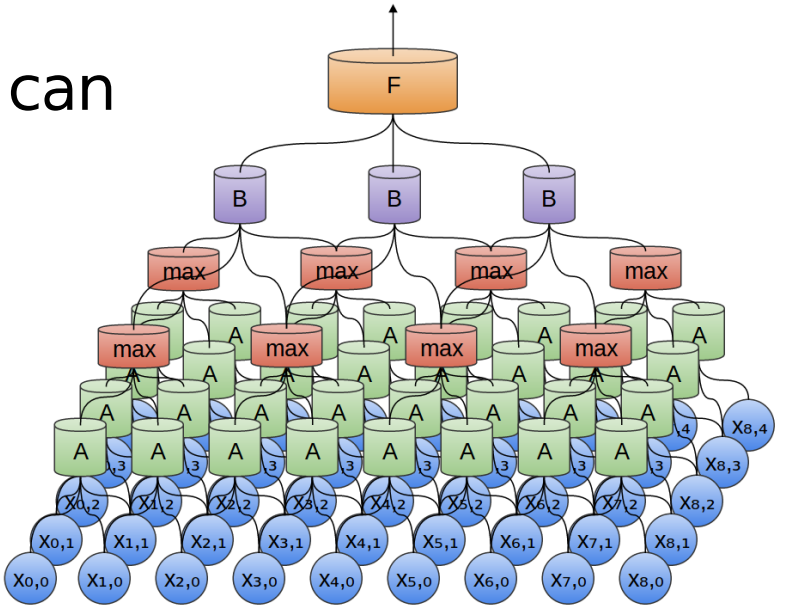
What is a Convolutional Neural Network?

- The signal can be 2D: the filters are now also 2D, but it's all essentially the same



What is a Convolutional Neural Network?

- The function computed by this gigantic model is **differentiable*** w.r.t. the weights
  - Given training data and a **loss function** measuring the deviation between predicted and actual values, we can optimize the weights by gradient descent
  - The gradient of the loss function can be found efficiently by a method called **back-propagation**



* nearly everywhere

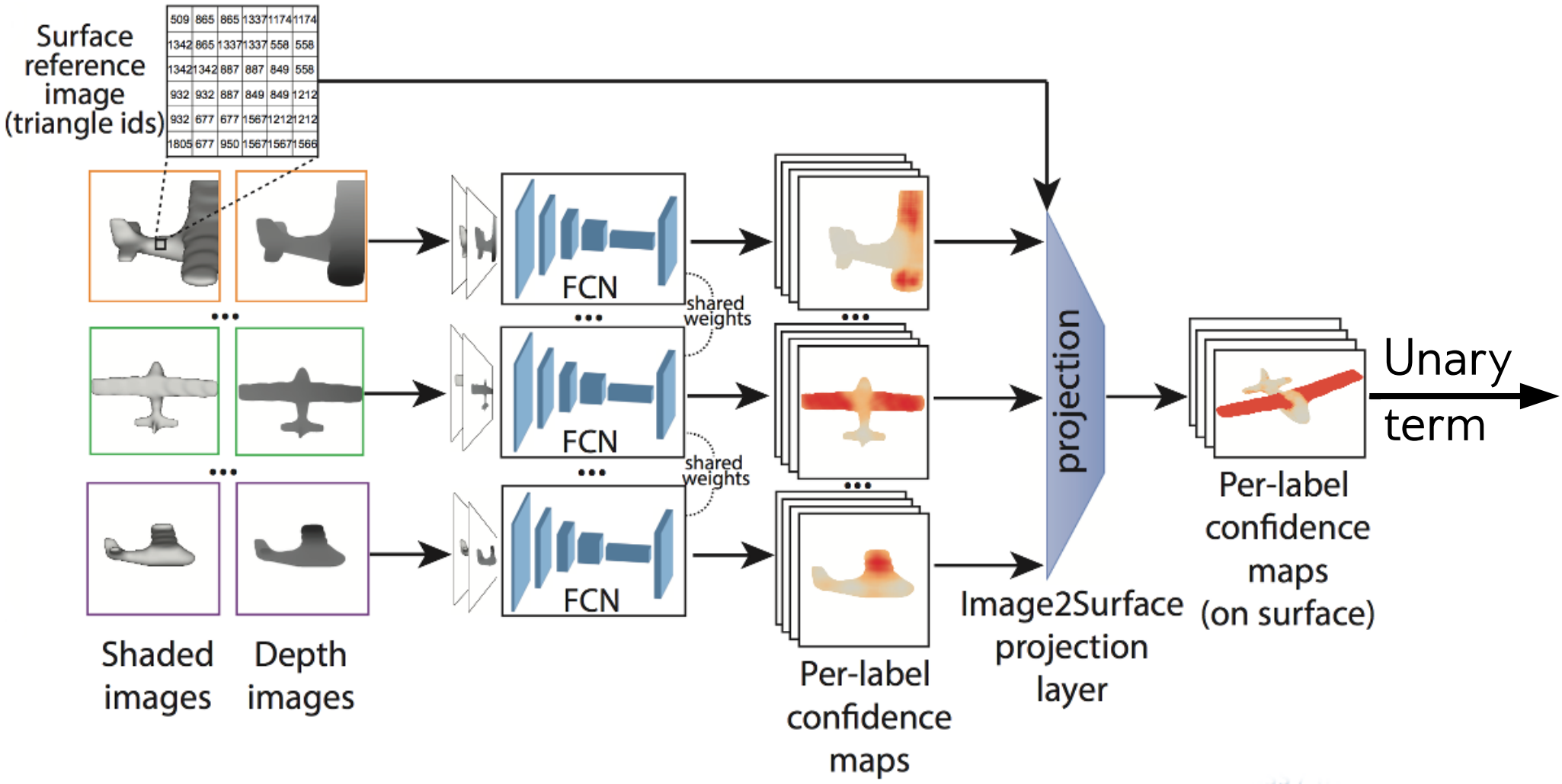
Kalogerakis, Averkiou, Maji, Chaudhuri, https://arxiv.org/abs/1612.02808

Applying 2D image CNNs to 3D shapes

- **Problem:** Doing convolutions on arbitrary surfaces at various scales is hard
- **Solution:** Render shape from different directions, apply image CNNs to the renderings, and merge the results
  - … also lets us take advantage of huge amounts of image training data to pre-train the model (voxel grids are typically 3D data-starved)

Kalogerakis, Averkiou, Maji, Chaudhuri, https://arxiv.org/abs/1612.02808

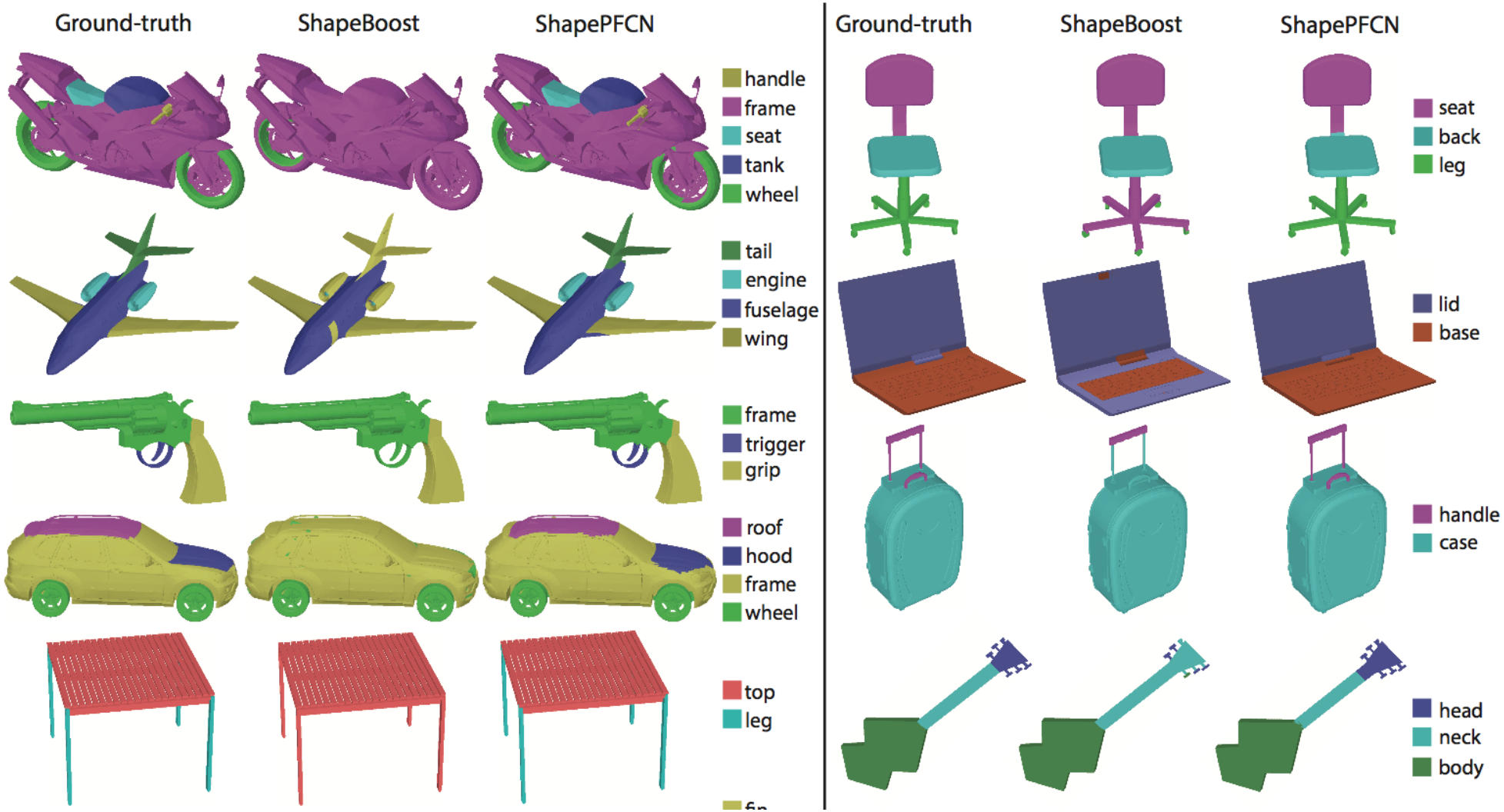
Applying 2D image CNNs to 3D shapes



This is a fully convolutional network that produces a probability for each label at each pixel. It does not predict the class of the overall object.

Kalogerakis, Averkiou, Maji, Chaudhuri, https://arxiv.org/abs/1612.02808

Results



Kalogerakis, Averkiou, Maji, Chaudhuri, https://arxiv.org/abs/1612.02808

Overall results (on ShapeNetCore)

	ShapeBoost	Guo et al.	ShapePFCN
Category Avg.	83.1	78.7	**88.7**
Category Avg. (>3 labels)	74.8	69.6	**84.9**
Dataset Avg.	80.4	74.7	**88.0**
Dataset Avg. (>3 labels)	74.2	68.7	**84.5**

Kalogerakis, Averkiou, Maji, Chaudhuri, https://arxiv.org/abs/1612.02808

Per-class results

	#train/test shapes	#part labels	ShapeBoost	Guo et al.	ShapePFCN
Airplane	250 / 250	4	84.1	78.4	**88.4**
Bag	38 / 38	2	94.3	**95.7**	95.5
Cap	27 / 28	2	**94.8**	91.2	92.0
Car	250 / 250	4	75.5	74.7	**86.6**
Chair	250 / 250	4	71.9	60.6	**83.7**
Earphone	34 / 35	3	76.0	74.6	**82.9**
Guitar	250 / 250	3	86.9	82.8	**89.7**
Knife	196 / 196	2	84.1	69.6	**87.1**
Lamp	250 / 250	4	63.8	57.7	**78.3**
Laptop	222 / 222	2	79.4	68.0	**95.2**
Motorbike	101 / 101	6	78.6	76.9	**87.5**
Mug	92 / 92	2	98.1	97.7	98.1
Pistol	137 / 138	3	84.9	82.9	**92.2**
Rocket	33 / 33	3	**83.2**	79.6	81.5
Skateboard	76 / 76	3	89.6	87.8	**92.5**
Table	250 / 250	3	83.9	81.0	**88.0**

Effect of solution components

	fixed views	disjoint training	unary term	use everything
Category Avg.	87.8	88.2	83.5	**88.7**
Category Avg. (>3 labels)	84.0	84.5	76.2	**84.9**
Dataset Avg.	87.2	87.5	82.2	**88.0**
Dataset Avg. (>3 labels)	83.6	84.2	76.7	**84.5**