3D Shape Representation

Tianqiang

04/01/2014

Why do we need 3D shapes?

- Image/video understanding
- Content creation

Why do we need 3D shapes?

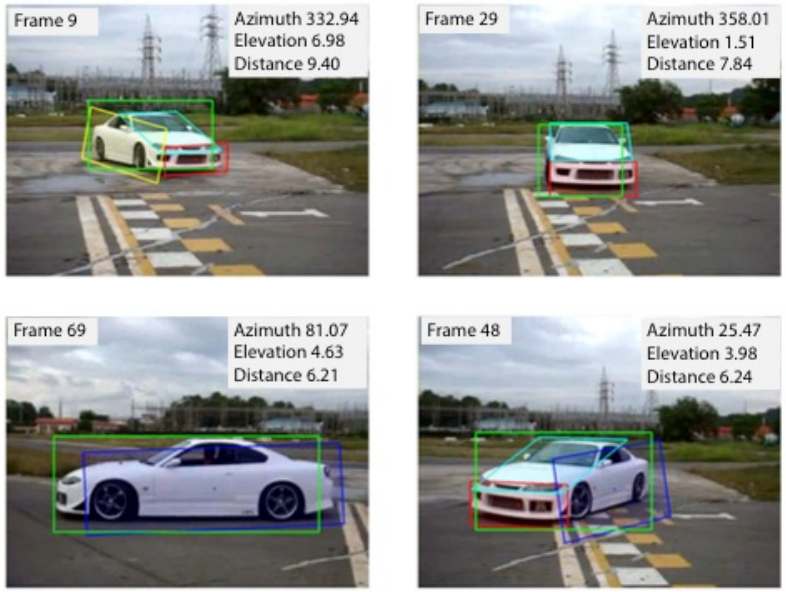
- Image/video understanding

[Chen et al. 2010]

Why do we need 3D shapes?

- Image/video understanding

[Chen et al. 2010]



[Xiang et al. 2014 (ECCV)]

Why do we need 3D shapes?

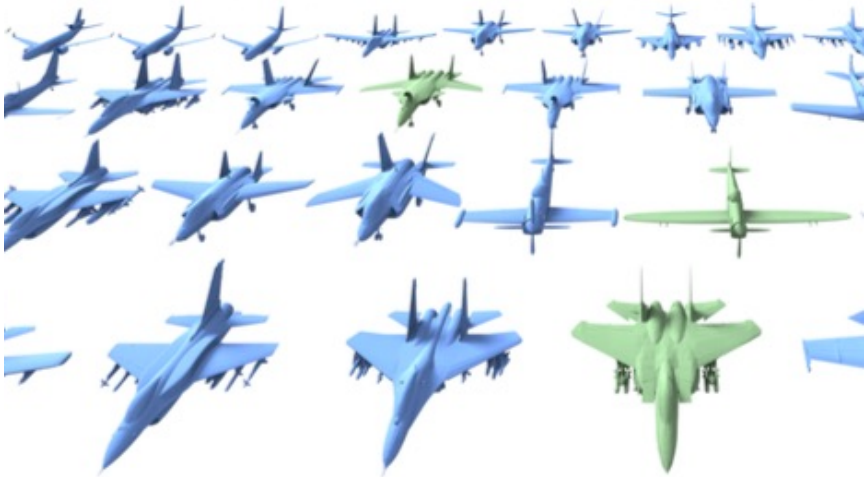
Image/video understanding

Need a 3D shape prior!

Why do we need 3D shapes?

- Image/video understanding
- Content creation

[Yumer et al. 2010]



[Kalogerakis et al. 2012]

Why do we need 3D shapes?

Content creation

Need a 3D shape prior, again!

Papers to be covered in this lecture

Recovering 3D from single images

- 3D Reconstruction from a Single Image [Rother et al. 08]
- Shared Shape Spaces [Prisacariu et al. 11]
- A Hypothesize and Bound Algorithm for Simultaneous Object Classification, Pose Estimation and 3D Reconstruction from a Single 2D Image [Rother et al. 11]
- Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild [Xiang et al. 14 (WACV)]
- Inferring 3D Shapes and Deformations from Single Views [Chen et al. 10]

Papers to be covered in this lecture

Tracking

- Simultaneous Monocular 2D Segmentation, 3D Pose Recovery and 3D Reconstruction [Prisacariu et al. 12]
- Monocular Multiview Object Tracking with 3D Aspect Parts [Xiang et al. 14 (ECCV)]

Papers to be covered in this lecture

Shape synthesis

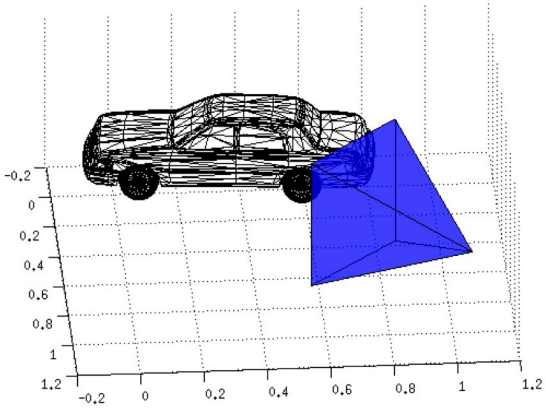
- A Probabilistic Model for Component-based Shape Synthesis [Kalogerakis et al. 12]
- Co-constrained Handles for Deformation in Shape Collections [Yumer et al. 14]

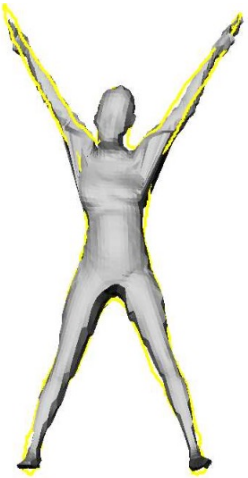
Recovering 3D from single images

Input: a single image

Recovering 3D from single images

Output: 3D shape, and pose or viewpoint





Outline

- A traditional generative model
- Shared shape space
- A benchmark for 3D object detection

Outline

- A traditional generative model
- Shared shape space
- A benchmark for 3D object detection

A Traditional Model

$$P(S,V \mid I) \propto P(S,V,I) = P(I \mid S,V)P(S)P(V)$$

S : 3D shape (output)

V : Viewpoint (output)

I : Image (input)

A Traditional Model

Projection likelihood Shape prior Viewpoint prior

$$P(S,V \mid I) \propto P(S,V,I) = \underline{P(I \mid S,V)}\,\underline{P(S)}\,\underline{P(V)}$$

S : 3D shape (output)

V : Viewpoint (output)

I : Image (input)

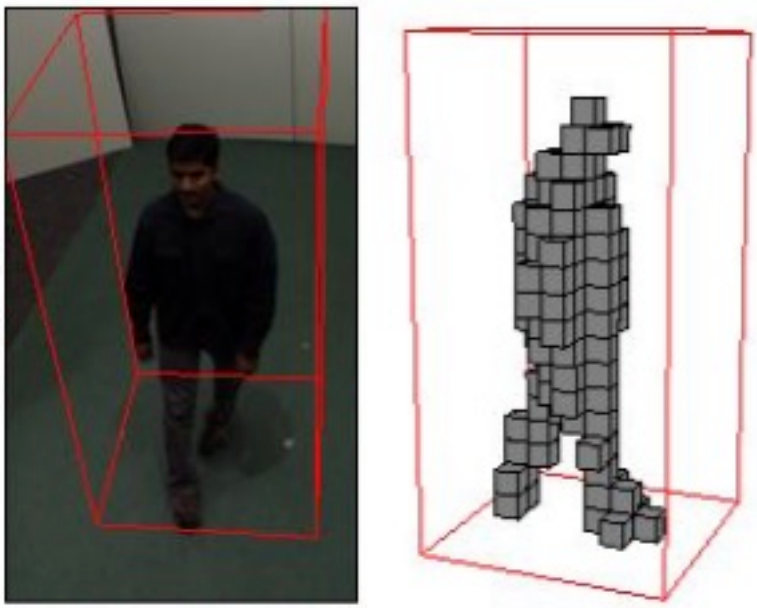
A Traditional Model

Projection likelihood Shape prior Viewpoint prior

$$P(S,V \mid I) \propto P(S,V,I) = \underline{P(I \mid S,V)}\,\underline{P(S)}\,\underline{P(V)}$$

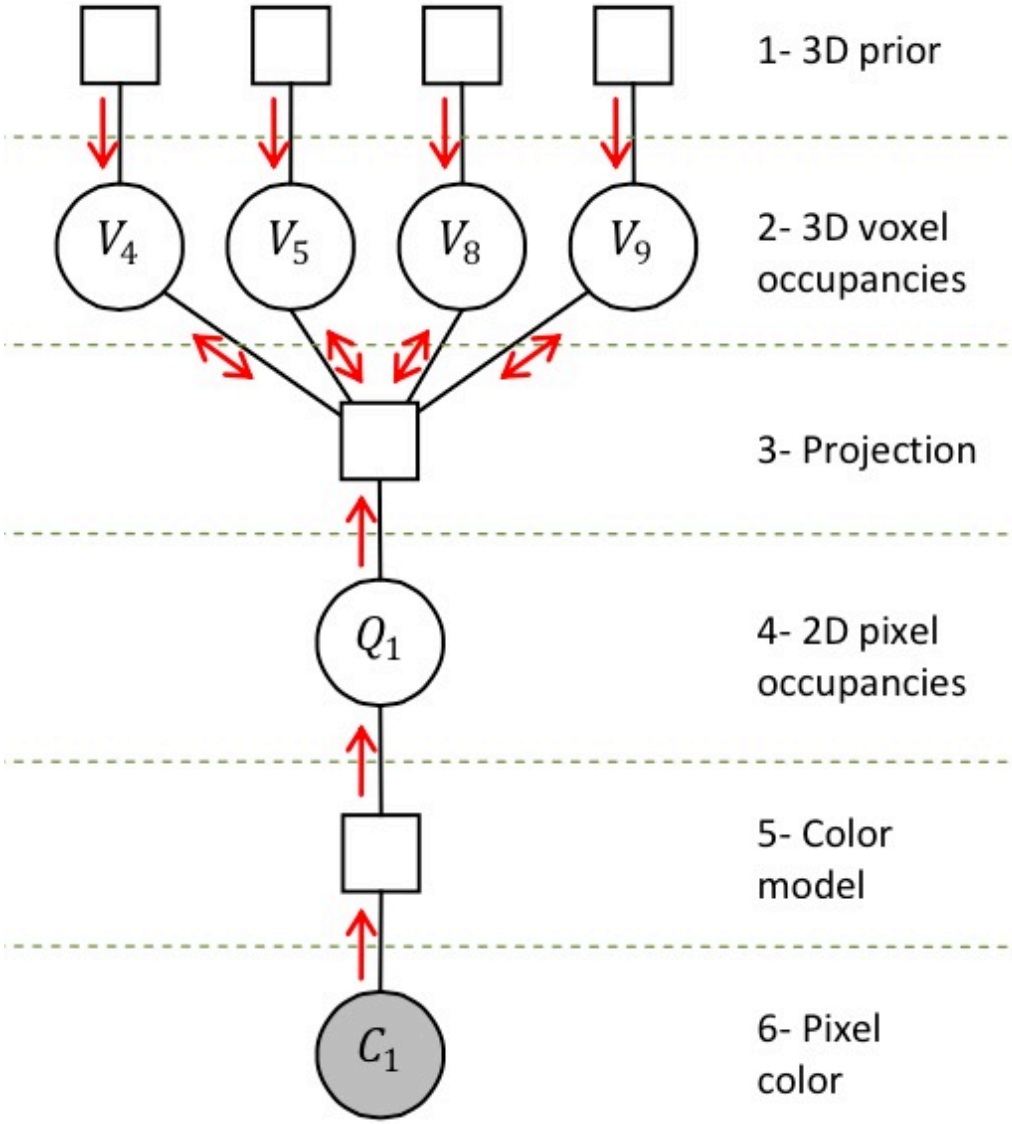
[Rother et al. 2008]
[Xiang et al. 2014]
[Chen et al. 2014]

Rother et al. 2008 - Overview



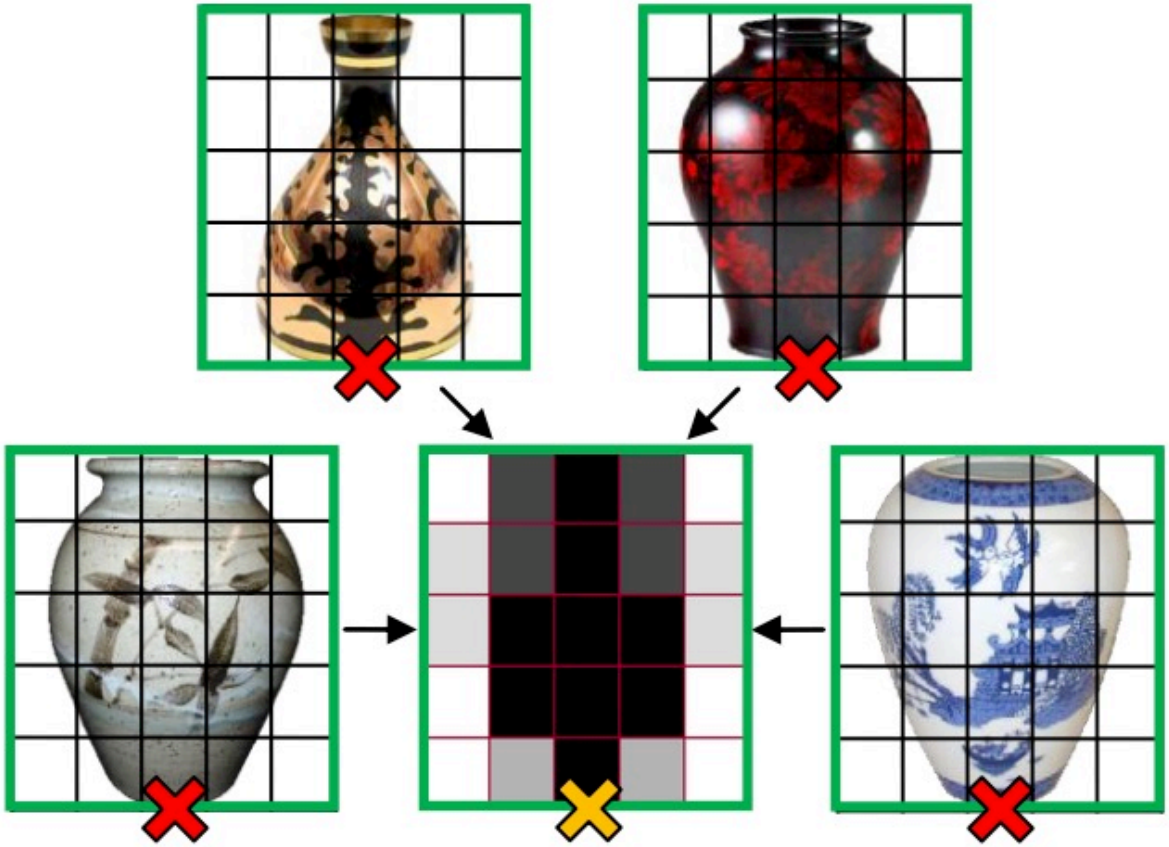
- Voxel representation for 3D shape
- Viewpoint is known or enumerated
- Using an optics law to do projection

Rother et al. 2008 - Overview



Rother et al. 2008 – 3D Shape Prior

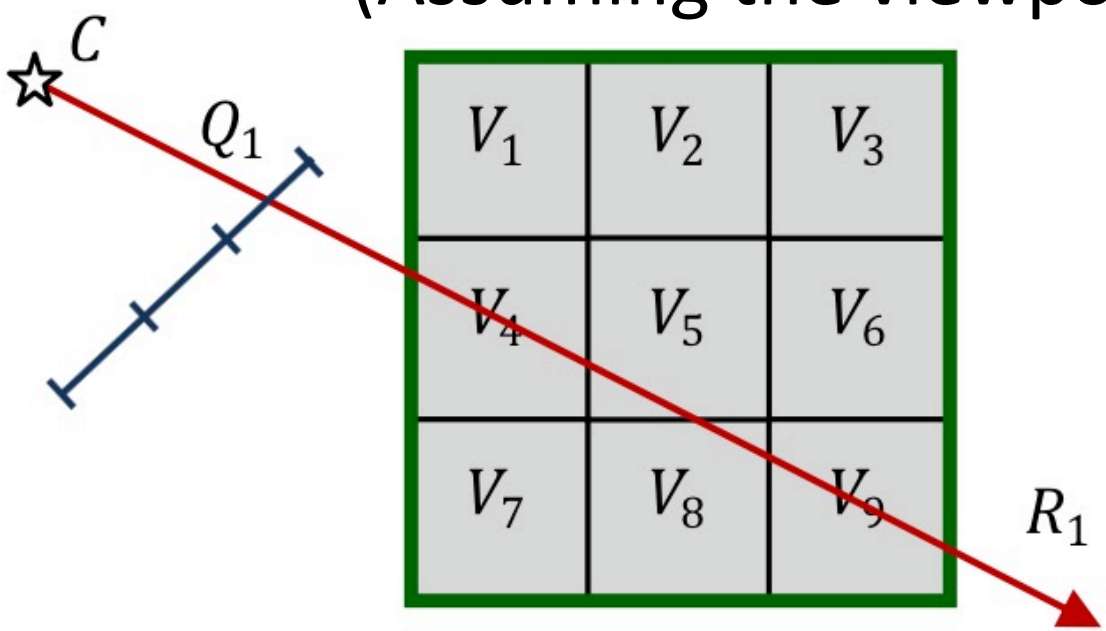
Voxel occupancies



(2D is just for illustration)

Rother et al. 2008 – Projection likelihood

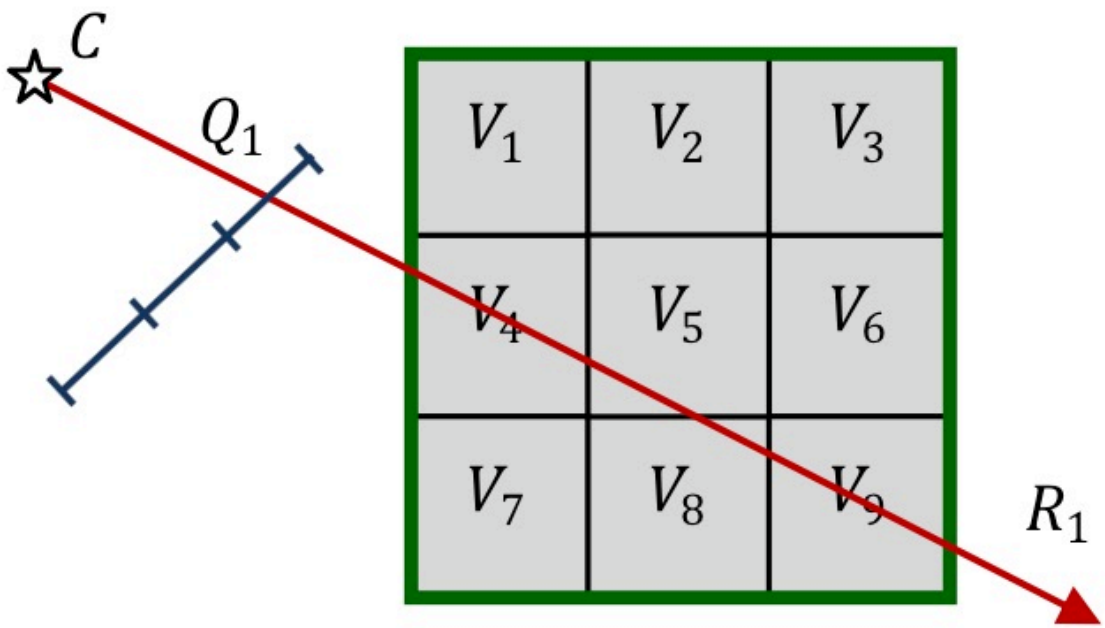
(Assuming the viewpoint is known)



$$\mathrm{E}(0) = 1 - \varepsilon_0$$
$$\mathrm{E}(1) = \varepsilon_1$$

$$P\left(Q_j = 0 \middle| V_1, \cdots, V_n\right) = \prod_{i=1}^{n} \mathrm{E}(V_i),$$

Rother et al. 2008 – Projection likelihood

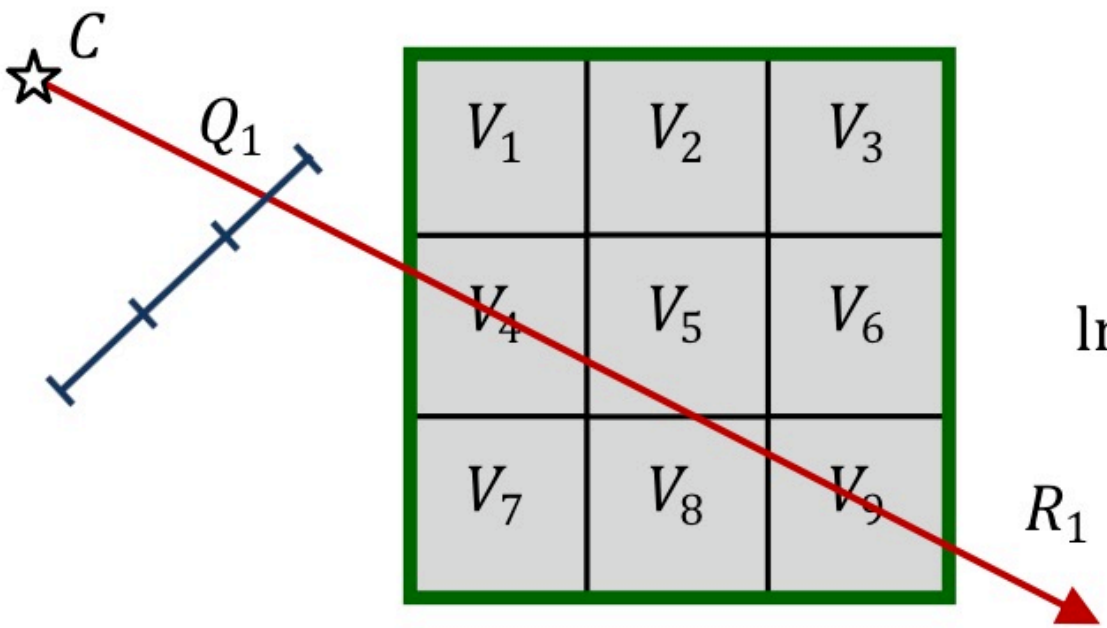


Two issues:

- Effect is unrelated to travel distance in each voxel
- Dependent on the resolution of grids

$$P\left(Q_j = 0 \middle| V_1, \cdots, V_n\right) = \prod_{i=1}^{n} \mathrm{E}(V_i),$$

Rother et al. 2008 – Projection likelihood



Beer-lambert law:

$$\frac{dI}{dr} = -I.\alpha.$$

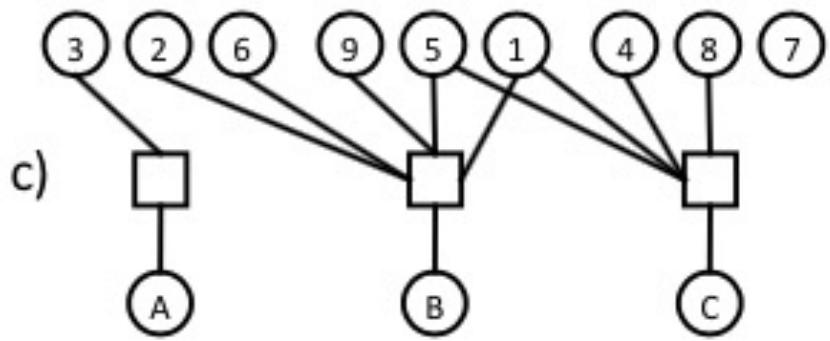
$$\ln\frac{I}{I_0} = -\alpha.r \Rightarrow P = \frac{I}{I_0} = e^{-\alpha.r}$$

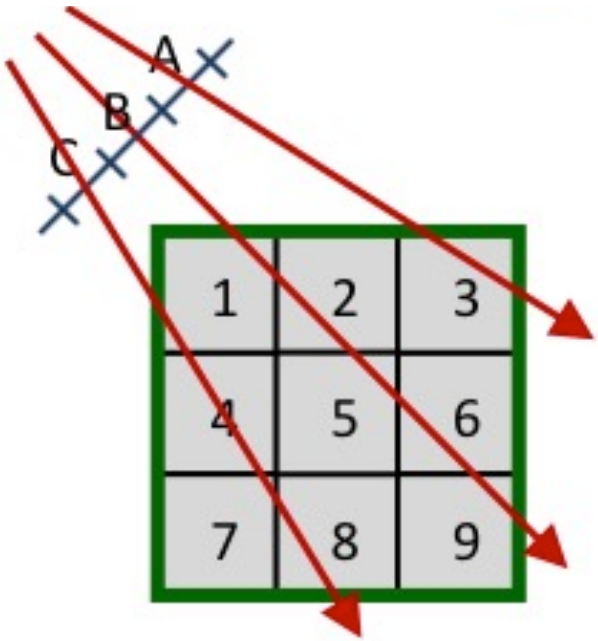
$$P\left(Q_j = 0 \middle| V_1, \cdots, V_n\right) = \prod_{i=1}^{n} e^{-\alpha(V_i).r_{ji}} = \prod_{i=1}^{n} \mathrm{E}(V_i)^{r_{ji}},$$

Rother et al. 2008 – Optimization

Possible solution: belief propagation

But what if the graph is loopy?



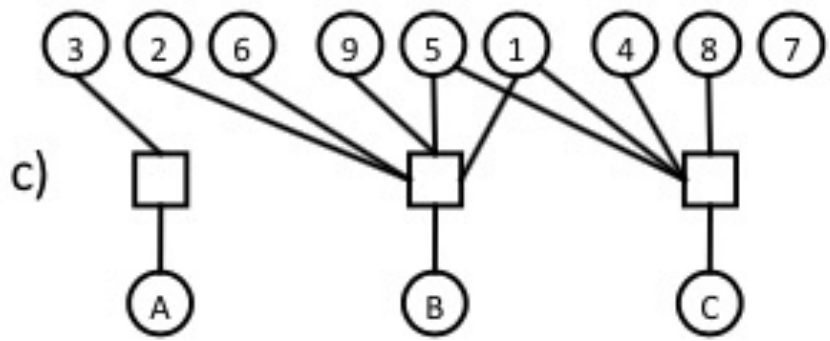


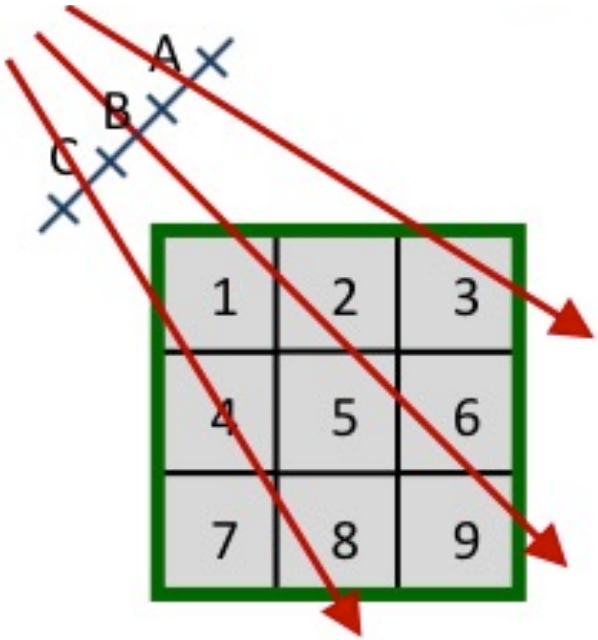
Rother et al. 2008 – Optimization

Possible solution: belief propagation

But what if the graph is loopy?

Down-sample the image!

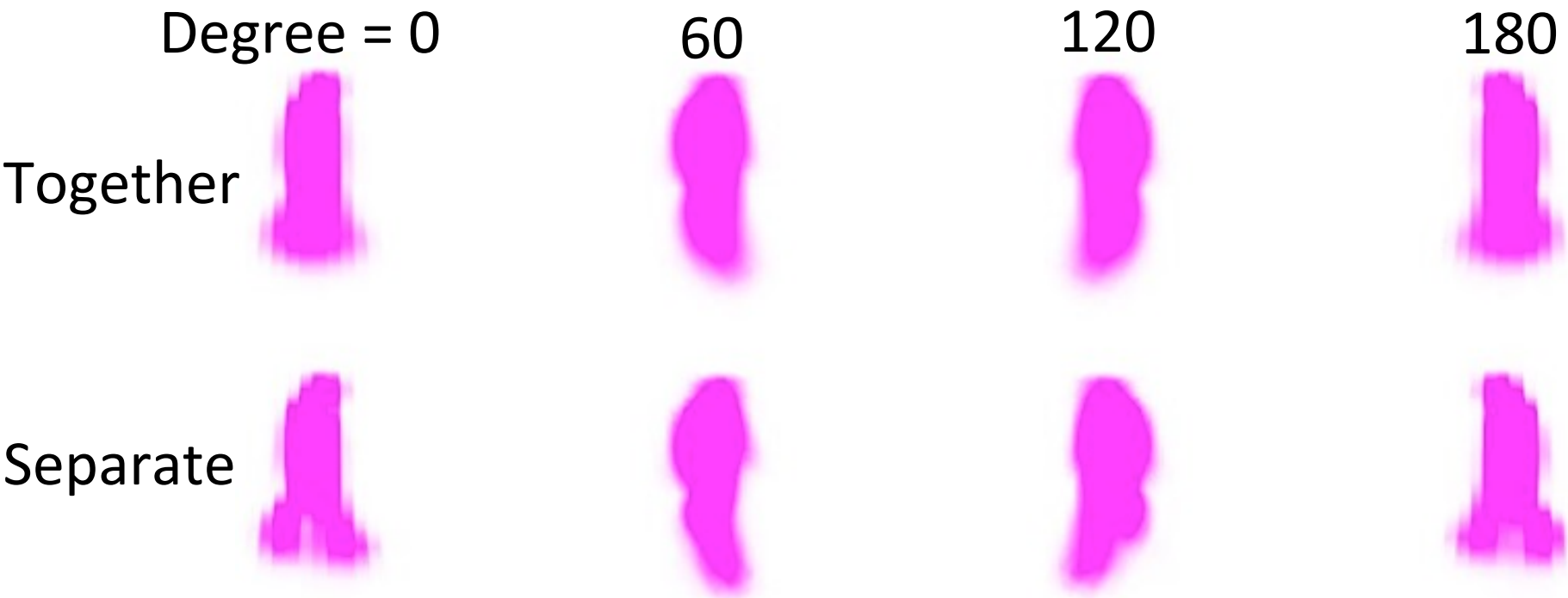




Rother et al. 2008 – Experiments

Learning shape prior: separately for each subclass

eg. Right leg in front, left leg up, …



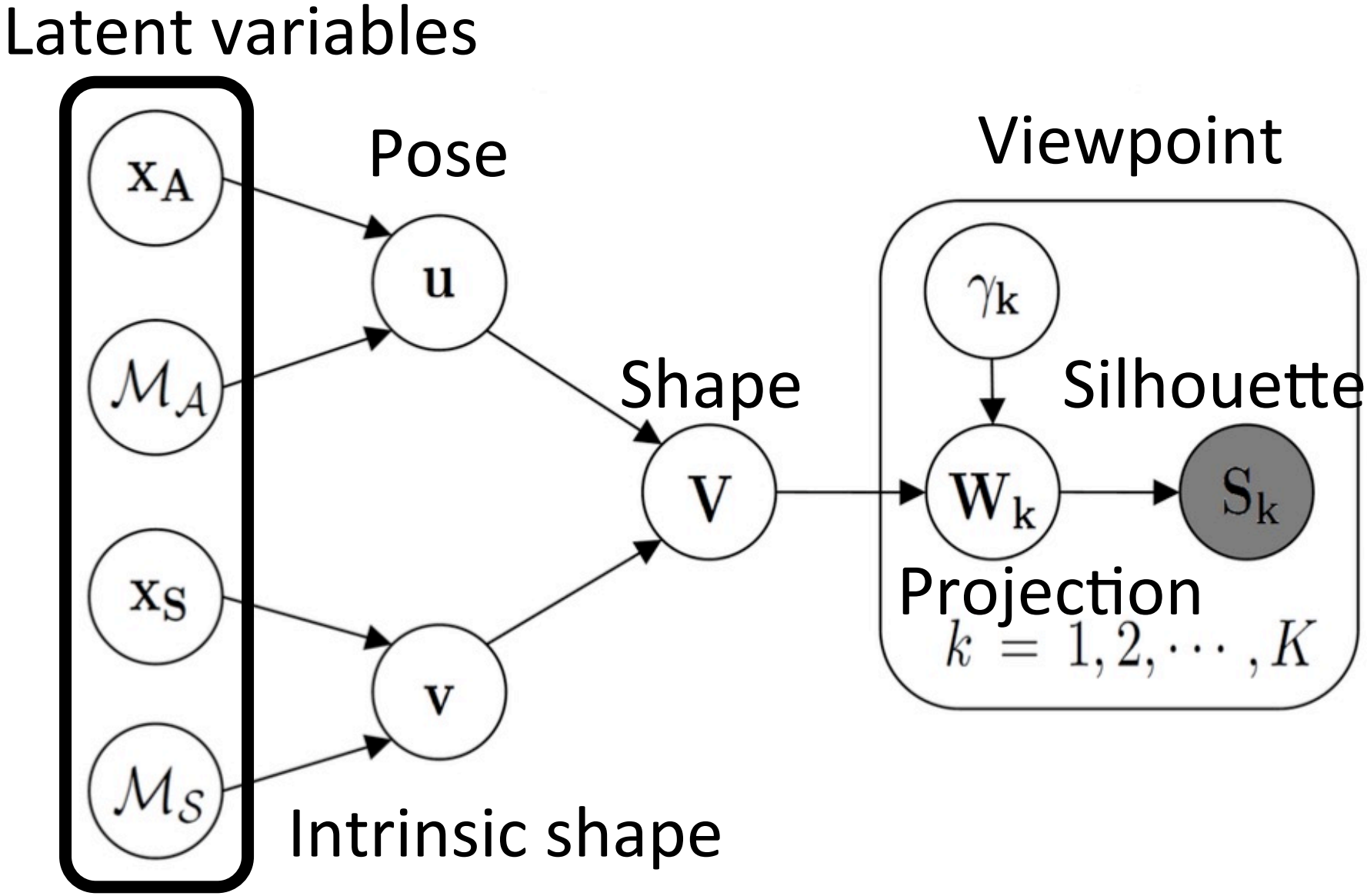
Rother et al. 2008 – Experiments

Unknown registration parameters	E_R (%)	V_R (%)	E_χ (cm)	E_φ (%)	E_θ (°)	E_ρ (cm)
-	37.1	-1.8	-	-	-	-
χ	37.2	-2.2	6.2	-	-	-
φ	37.8	-2.2	-	57	-	-
θ	37.5	-2.3	-	-	45	-
ρ	36.4	-7.4	-	-	-	5.0
χ, φ	37.7	-2.9	8.5	48	-	-
χ, φ, θ	38.1	-3.6	12.4	81	65	-

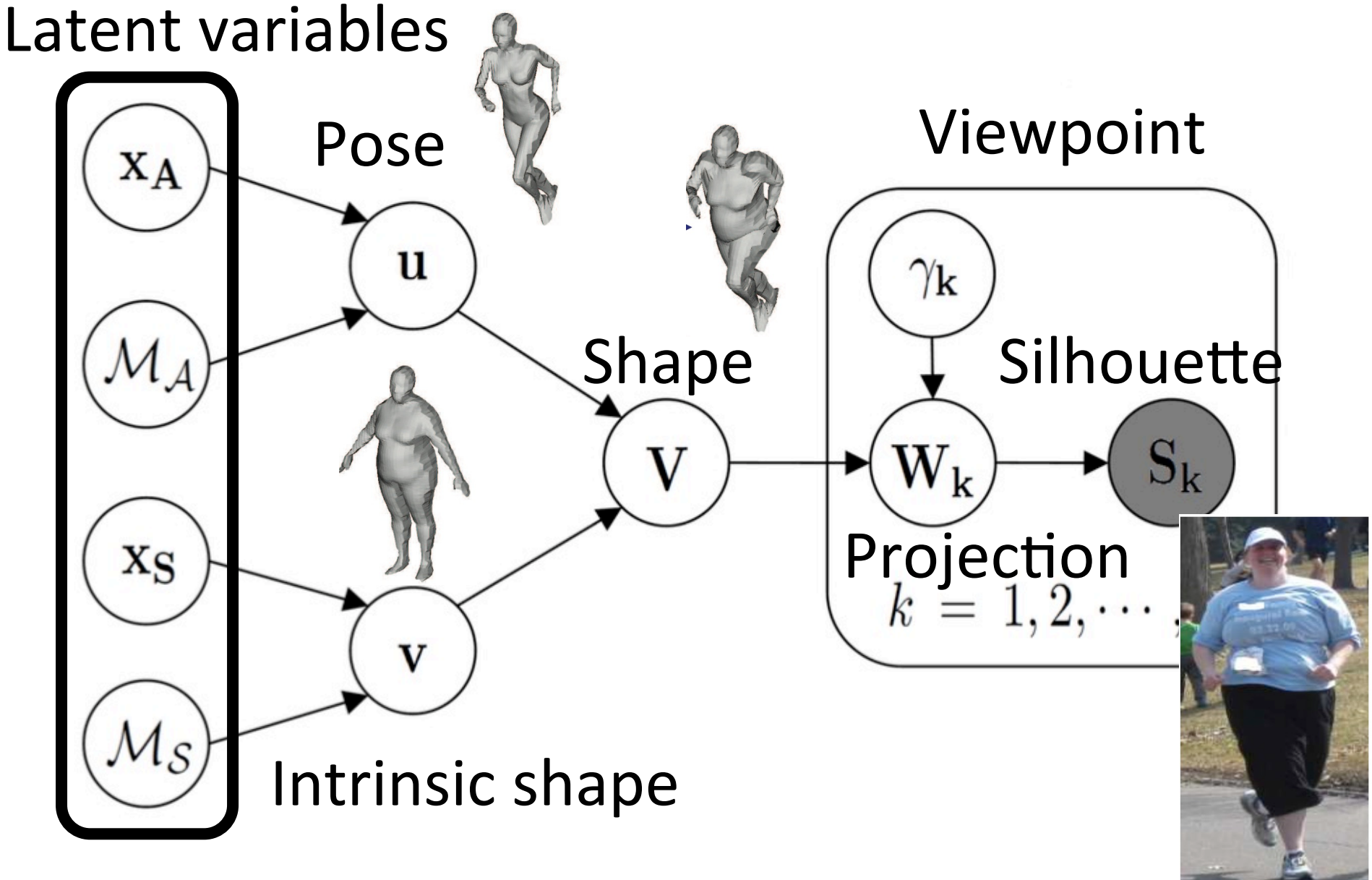
Chen et al. 2010 - Overview

- Modeling intrinsic shape (phenotype) and pose variations
- Mesh representation for 3D shape.
- Only leveraging silhouettes in images

Chen et al. 2010 - Overview

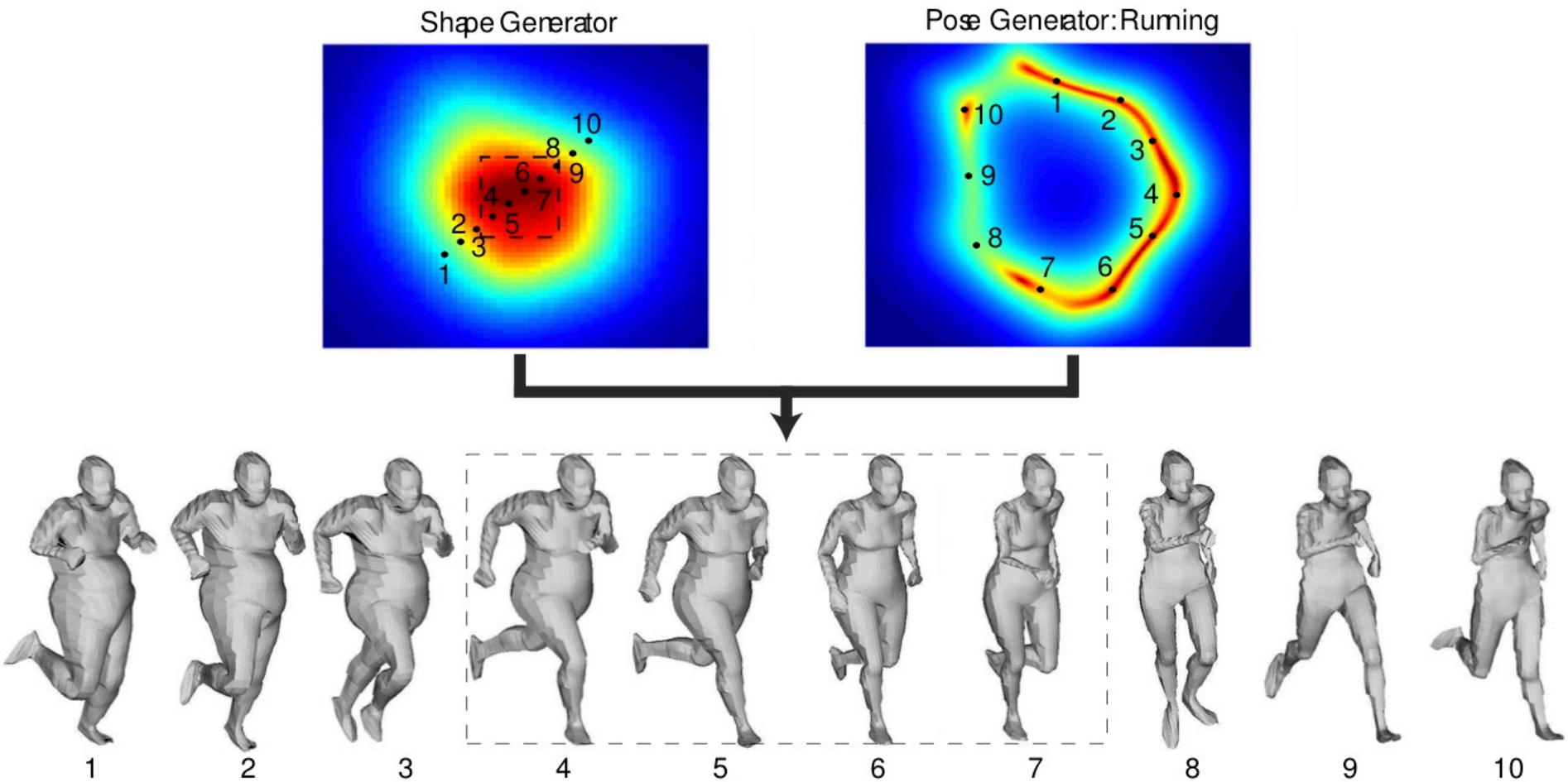


Chen et al. 2010 - Overview



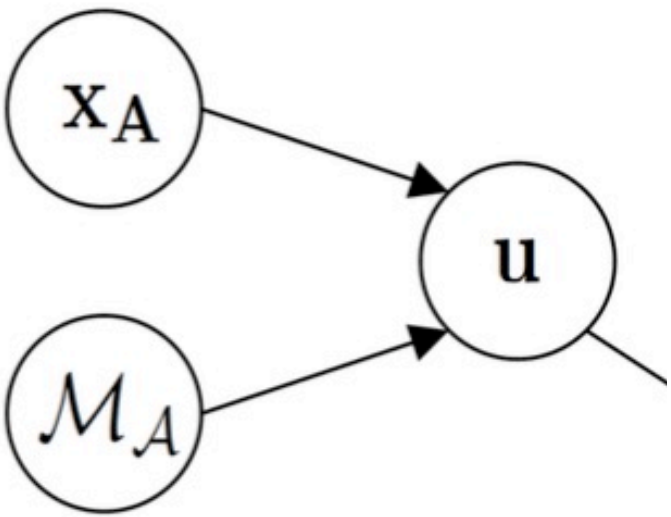
Chen et al. 2010 – 3D Shape Prior

Gaussian Process Latent Variable Models



Gaussian Process Latent Variable Model

Pose

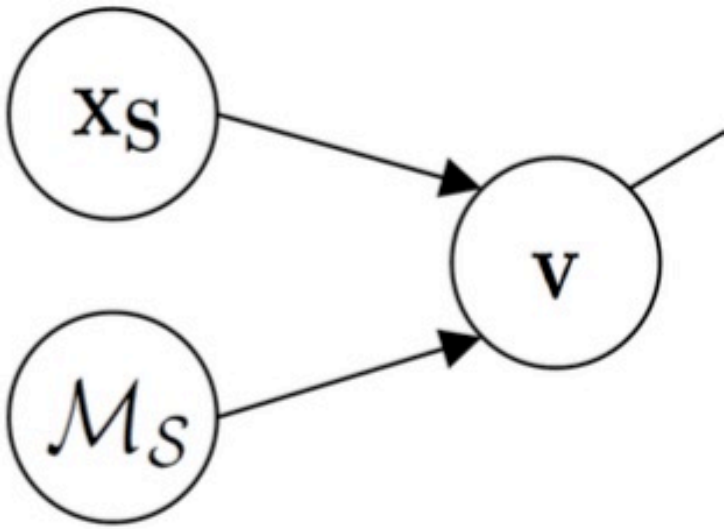


$$P(\mathbf{u}|\mathbf{x_A}, \mathcal{M}_\mathcal{A})$$

$$= \mathcal{N}\left(\mathbf{u}; \mathbf{k_U^T}(\mathbf{x_A})\mathbf{K_U^{-1}}\mathbf{Y_A}, (k_U(\mathbf{x_A}, \mathbf{x_A}) - \mathbf{k_U^T}(\mathbf{x_A})\mathbf{K_U^{-1}}\mathbf{k_U}(\mathbf{x_A}))\mathbf{I}\right)$$
$$= N\left(\mathbf{u}; \bar{\mathbf{u}}(\mathbf{x_A}), \sigma_A^2(\mathbf{x_A})\mathbf{I}\right) \tag{2}$$

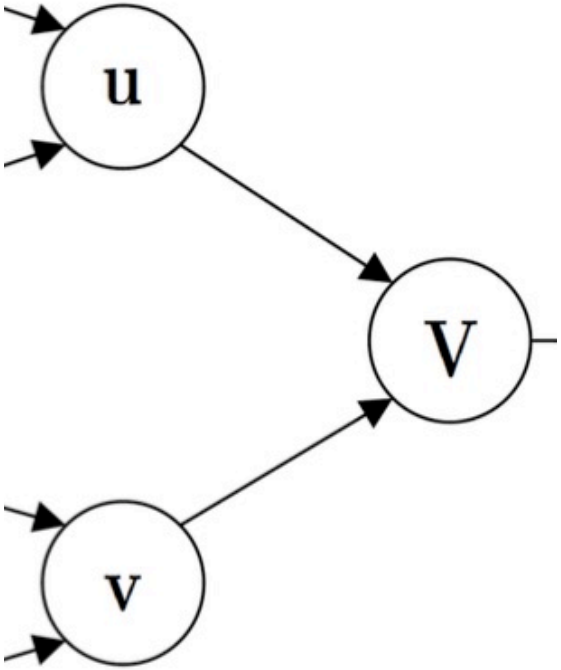
Gaussian Process Latent Variable Model

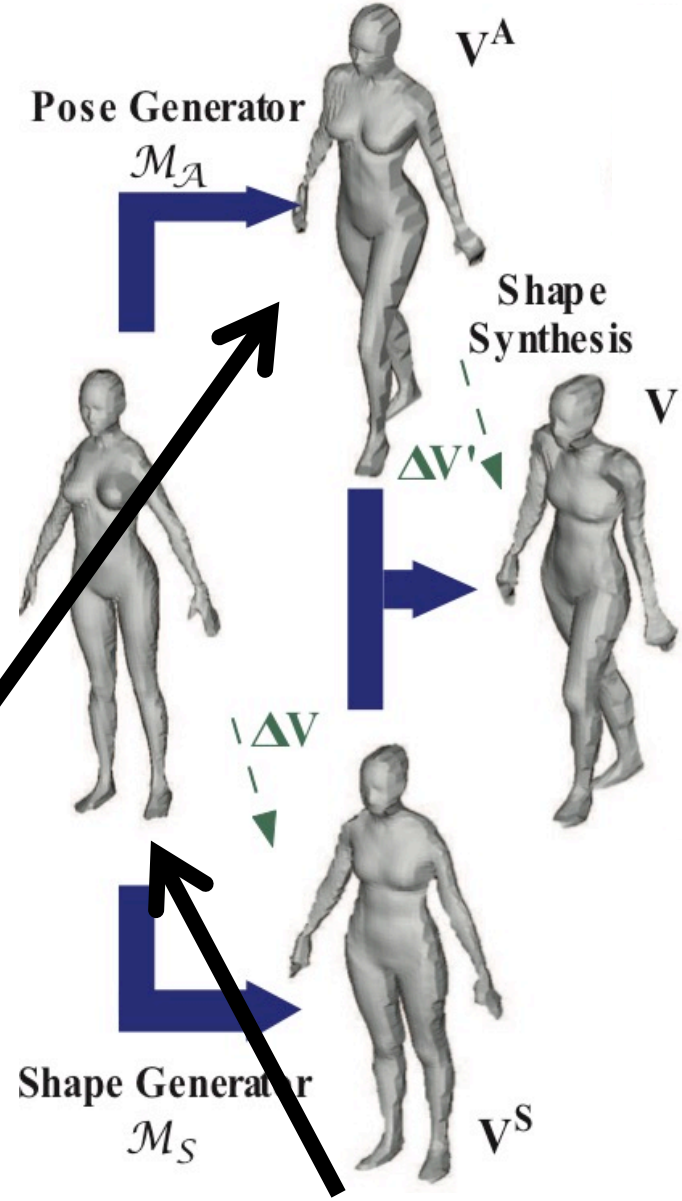
Intrinsic shape (phenotype)



$$
\begin{aligned}
&P(\mathbf{v}|\mathbf{x_S}, \mathcal{M}_S) \\
&= \mathcal{N}\left(\mathbf{v}; \mathbf{k_V^T}(\mathbf{x_S})\mathbf{K_V^{-1}}\mathbf{Y_S}, (k_V(\mathbf{x_S}, \mathbf{x_S}) - \mathbf{k_V^T}(\mathbf{x_S})\mathbf{K_V^{-1}}\mathbf{k_V}(\mathbf{x_S}))\mathbf{I}\right) \\
&= N\left(\mathbf{v}; \bar{\mathbf{v}}(\mathbf{x_S}), \sigma_S^2(\mathbf{x_S})\mathbf{I}\right).
\end{aligned}
$$

Shape transfer



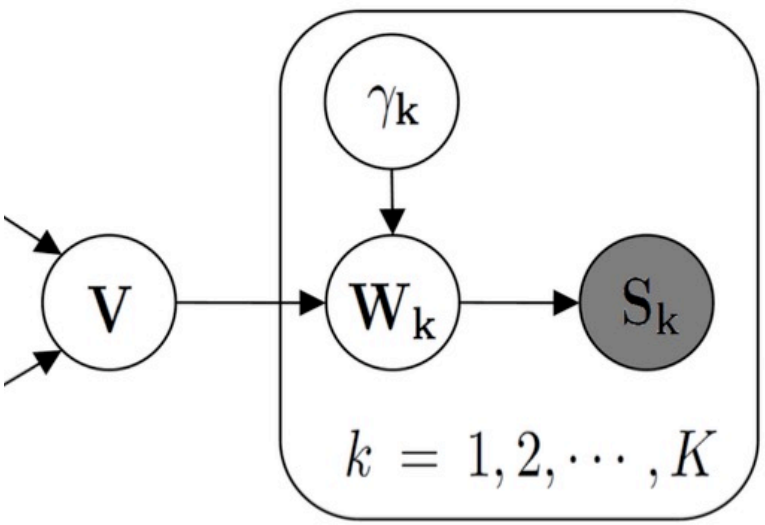


$$\Delta \mathbf{V}_\mathbf{i}' = \mathbf{J}_\mathbf{i} \Delta \mathbf{V}_\mathbf{i}.$$

$$\mathbf{J}_\mathbf{i} = [\mathbf{m}_\mathbf{i}^{\mathbf{A},1}, \mathbf{m}_\mathbf{i}^{\mathbf{A},2}, \mathbf{m}_\mathbf{i}^{\mathbf{A},3}][\mathbf{m}_\mathbf{i}^{\mathbf{O},1}, \mathbf{m}_\mathbf{i}^{\mathbf{O},2}, \mathbf{m}_\mathbf{i}^{\mathbf{O},3}]^{-1}$$

Chen et al. 2010 – Projection

$$P(S_k \mid V, \gamma_k) = P(W_k \mid V, \gamma_k) P(S_k \mid W_k)$$



S_k : input silhouettes

W_k : projected silhouettes

V : samples on 3D mesh

γ_k : camera parameters

Chen et al. 2010 – Projection

$$P(S_k \mid V, \gamma_k) = \underbrace{P(W_k \mid V, \gamma_k)}_{\text{Projecting}} \underbrace{P(S_k \mid W_k)}_{\text{Matching}}$$

Chen et al. 2010 – Projection

Projecting:

$$P(\mathbf{W_k}|\mathbf{V}, \gamma_\mathbf{k}) = \mathcal{N}(\mathbf{W_k}; \tilde{\mathbf{P}}_\mathbf{k}\mathbf{V} + \tilde{\mathbf{t}}_k, \sigma_w^2\mathbf{I}),$$

$\tilde{P}_k$: projection

$\tilde{t}_k$: translation vector

Chen et al. 2010 – Projection

Matching:

$$P(\mathbf{S_k}|\mathbf{W_k}) = \frac{1}{Z}\exp\left(-\frac{1}{2\sigma_s^2}DT^2_{\mathbf{S_k}}(\mathbf{W_k}))\right)$$

DT : a similarity function

Z : normalizer

Chen et al. 2010 – Optimization

Approximation

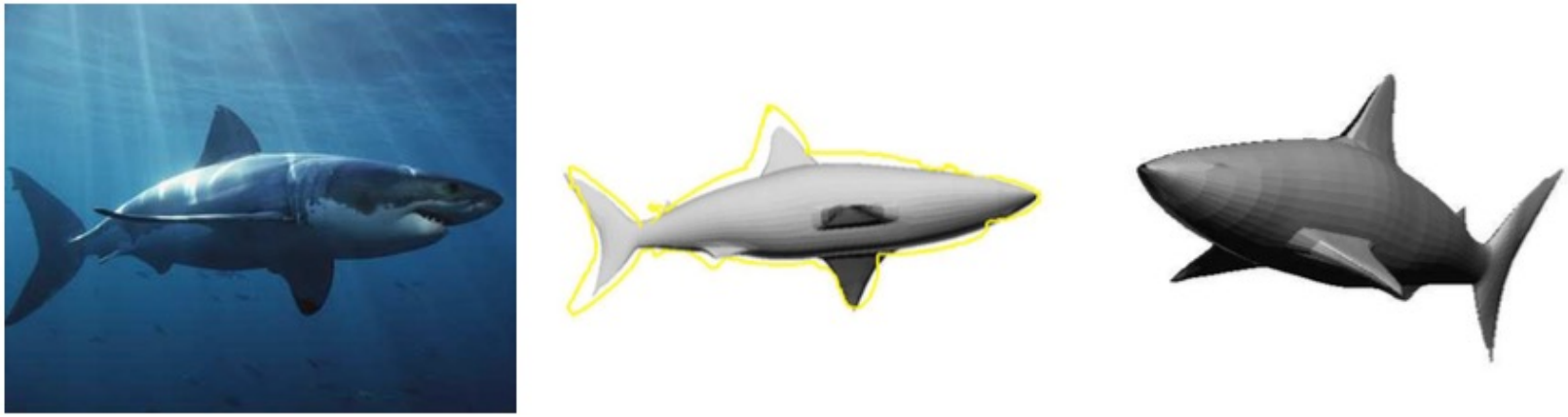
$$P(\{\mathbf{S_k}\}_{k=1}^{K}|\mathbf{x_A}, \mathbf{x_S}, \mathcal{M}_{\mathcal{A}}, \mathcal{M}_{\mathcal{S}}, \{\gamma_\mathbf{k}\}_{k=1}^{K}) \geq Q(\mathbf{x_A}, \mathbf{x_S}, \{\gamma_\mathbf{k}\}_{k=1}^{K})$$

$$= \prod_{k=1}^{K} \frac{1}{Z_k \sqrt{\det\left(\mathbf{I} + \frac{1}{\sigma_s^2}\mathbf{\Sigma_{W_k}}\right)}} \exp\left(-\frac{1}{2\sigma_s^2} DT^2_{\mathbf{S_k}}\left(\mu_{\mathbf{W_k}}\right)\right).$$

$$(\mathbf{x_A^{ML}}, \mathbf{x_S^{ML}}, \{\gamma_\mathbf{k}^{\mathbf{ML}}\}_{k=1}^{K}) \approx \arg \min_{\mathbf{x_A}, \mathbf{x_S}, \{\gamma_\mathbf{k}\}_{k=1}^{K}} -\log Q(\mathbf{x_A}, \mathbf{x_S}, \{\gamma_\mathbf{k}\}_{k=1}^{K}).$$

- Adaptive-scale line search
- Multiple initialization

Chen et al. 2010 – Experiments



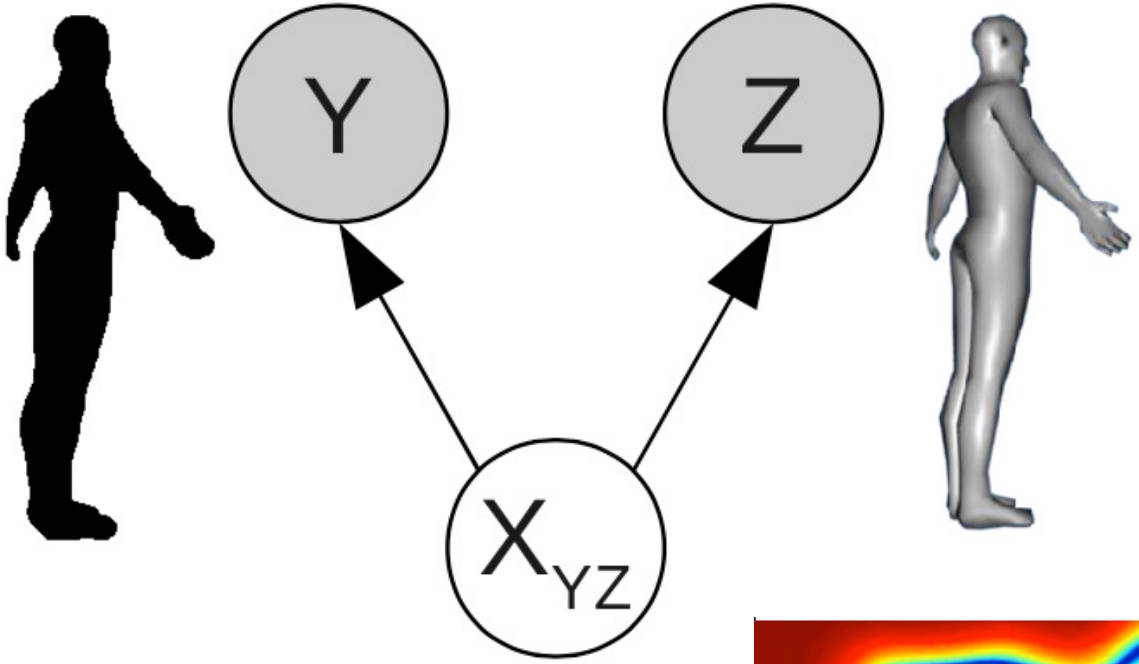
Data	Precision	Recall
22 sharks	0.8996 ± 0.0481	0.9308 ± 0.0380
20 human bodies	0.7801 ± 0.0689	0.8952 ± 0.0995

Outline

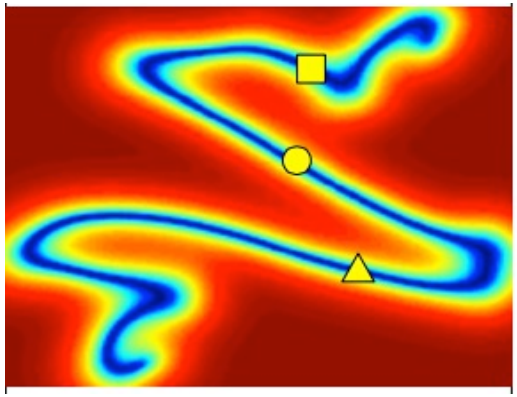
- A traditional generative model
- Shared shape space
- A benchmark for 3D object detection

Shared Shape Space

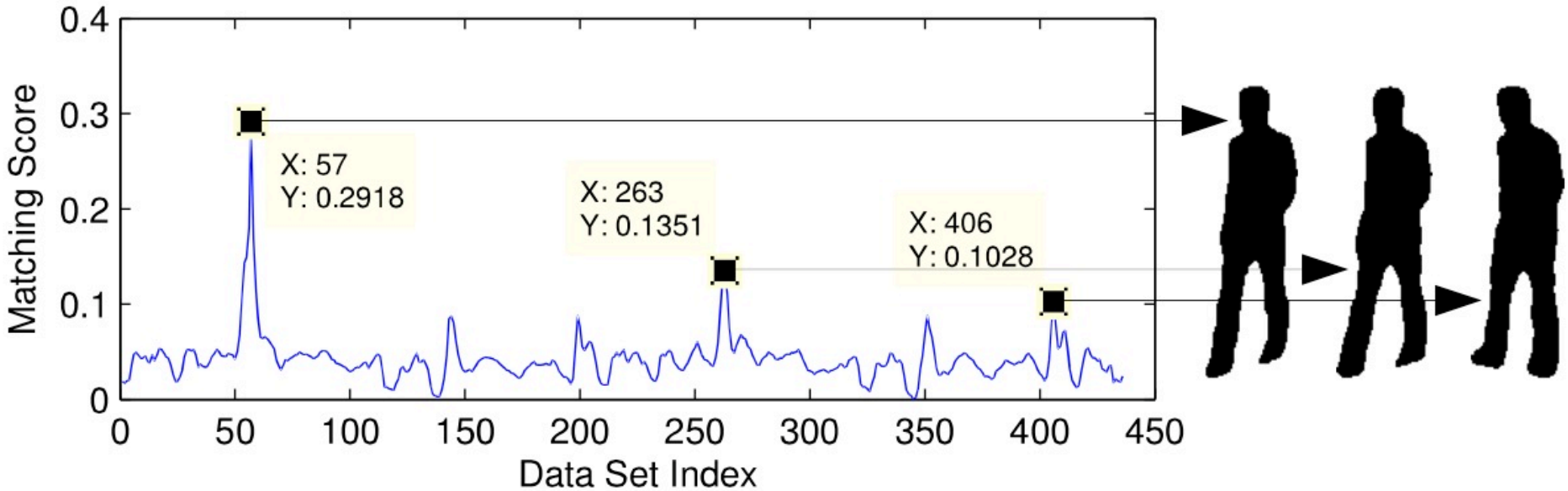
Silhouette 3D shape



Latent variable



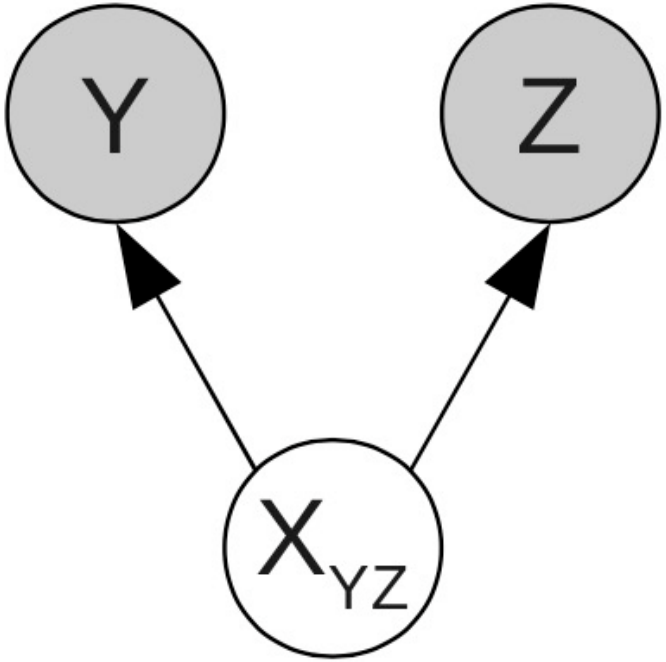
Shared Shape Space



Multimodality

Shared Shape Space

Silhouette 3D shape

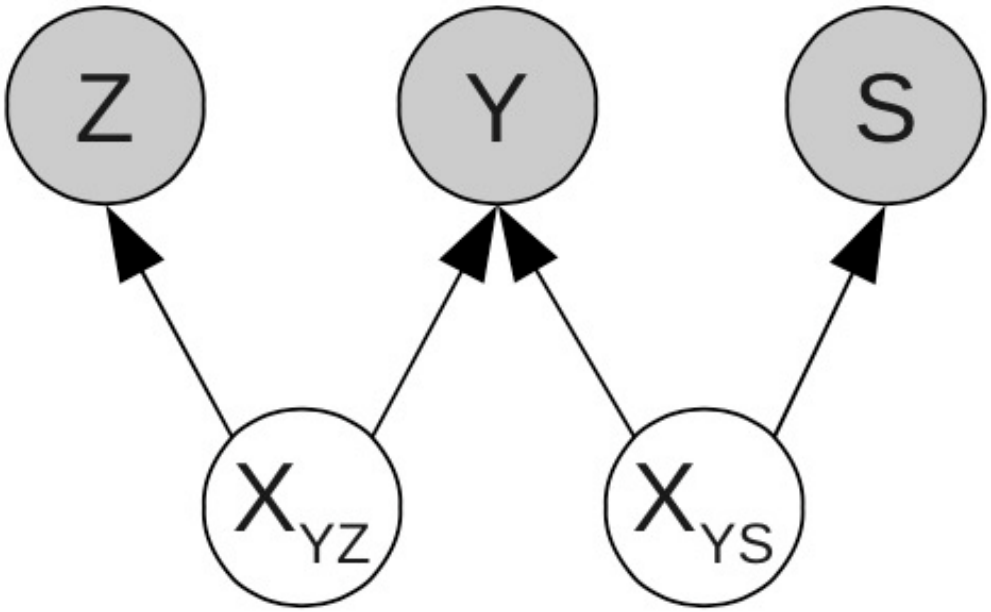


Latent variable

Cannot model multimodality

Shared Shape Space

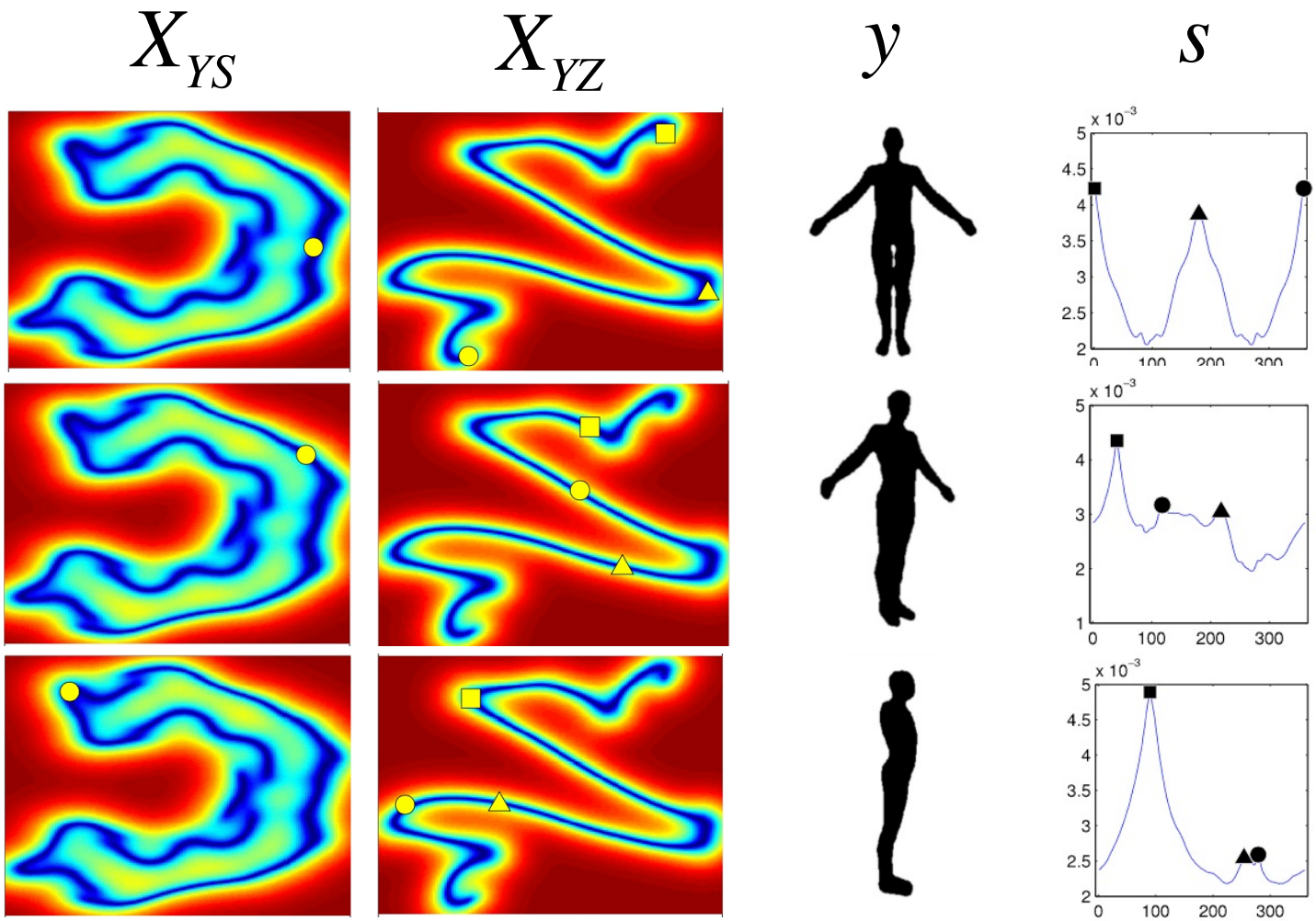
3D shape Silhouette Similarity distribution



Can model multimodality

Latent variables

Shared Shape Space

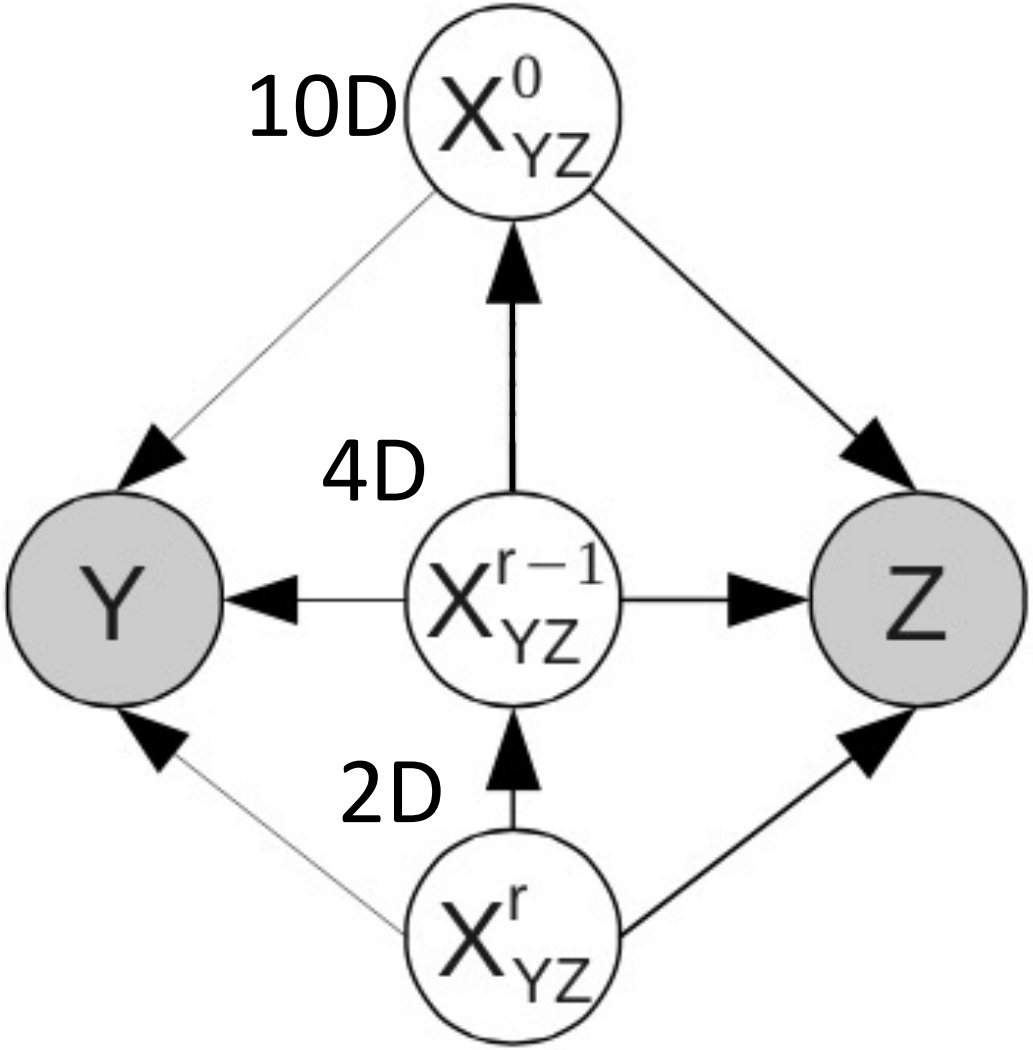


Shared Shape Space

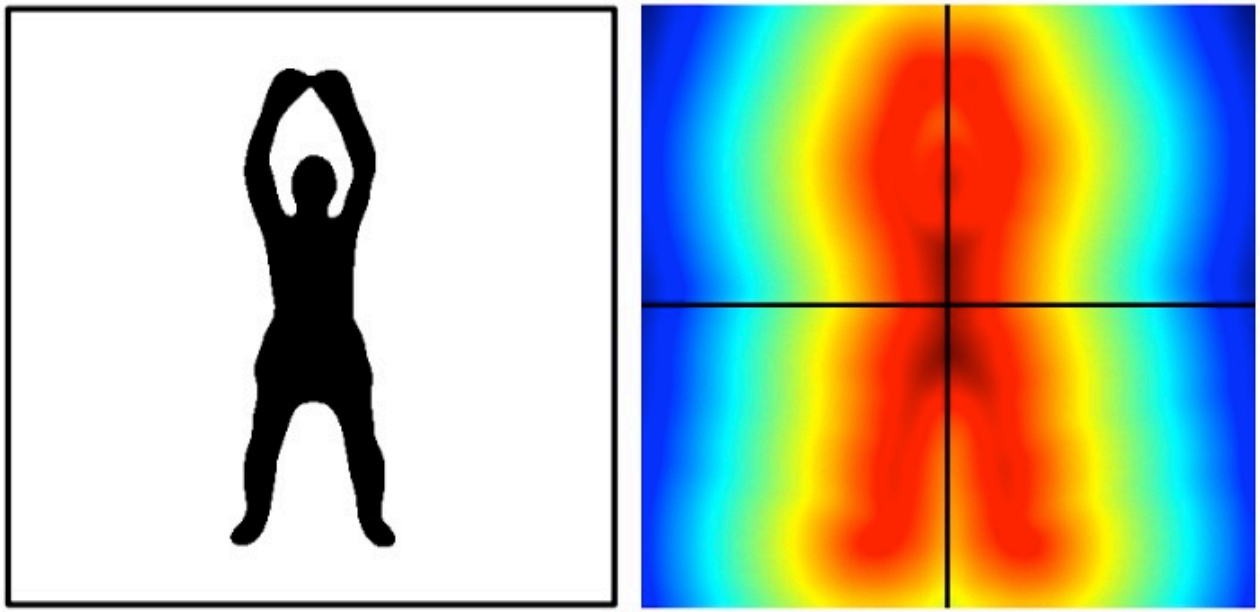
A little more detail:

- Hierarchy of latent manifolds
- Representation of 2D and 3D shapes

Hierarchy of Manifolds



2D/3D Discrete Cosine Transforms



Advantages:

- Fewer dimensions than directly representing the images.

2D/3D Discrete Cosine Transforms

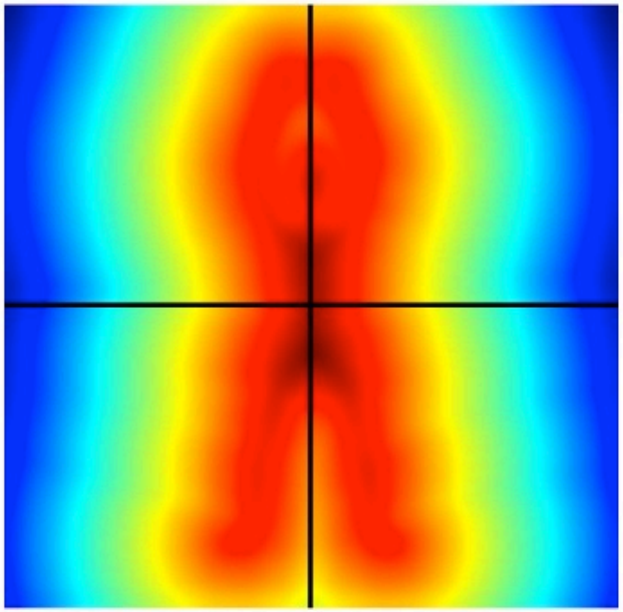
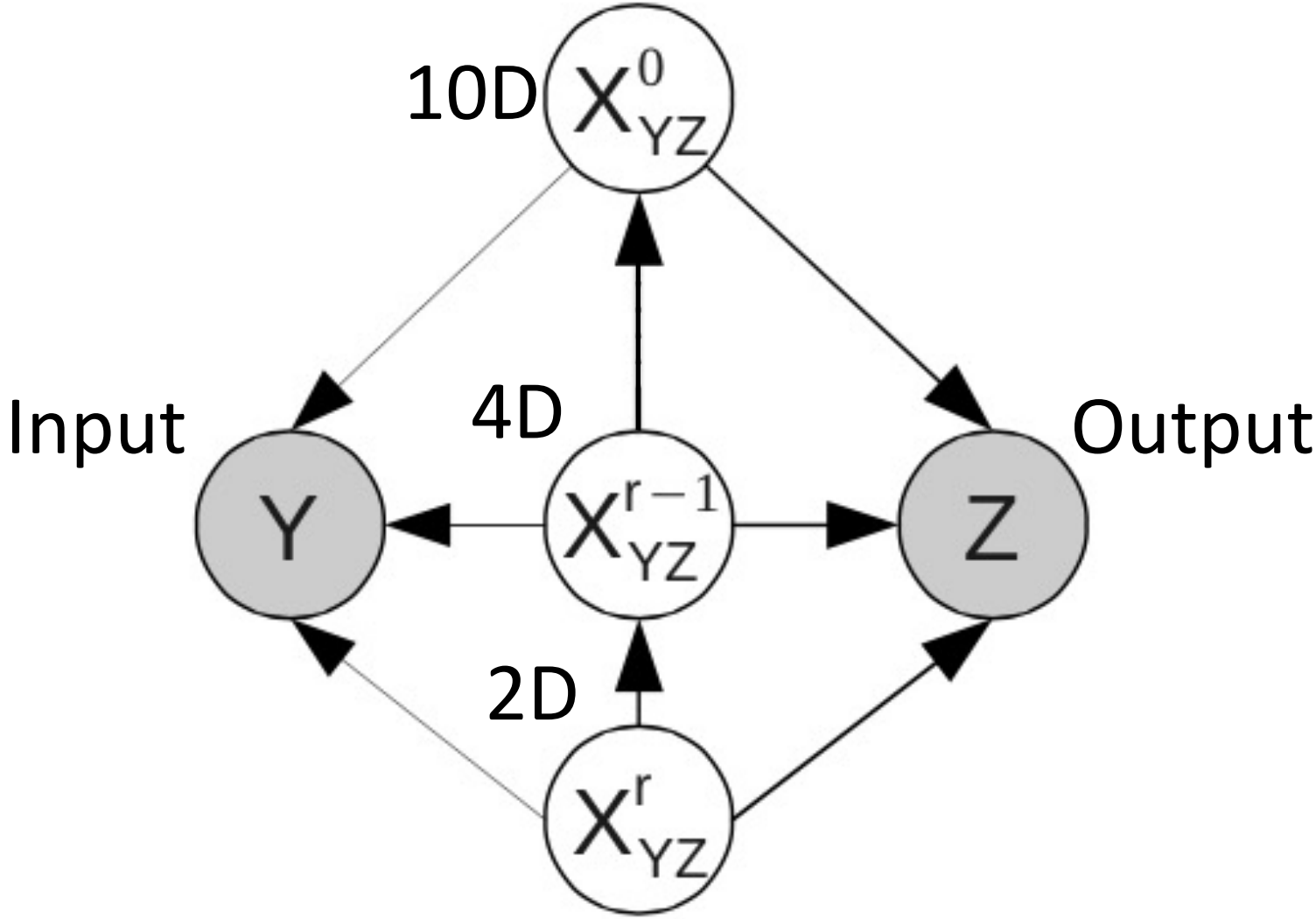


Image:
640X480 = 307200 dims

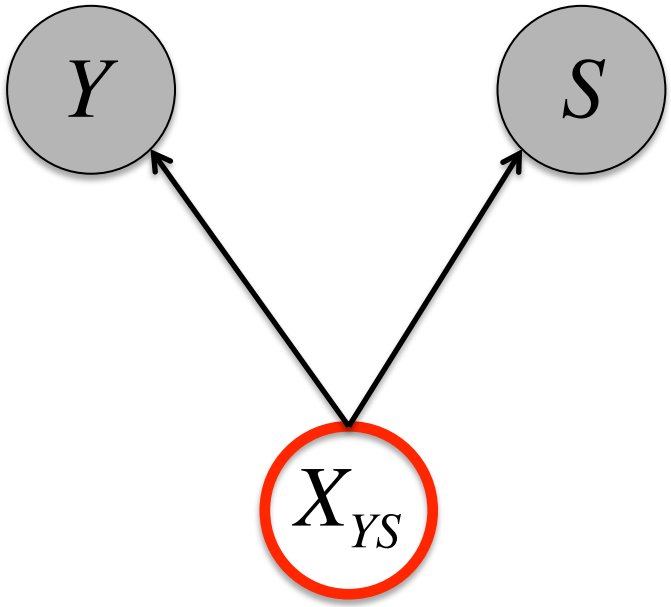
DCT:
35X35 = 1225 dims

Shared Shape Space - Inference



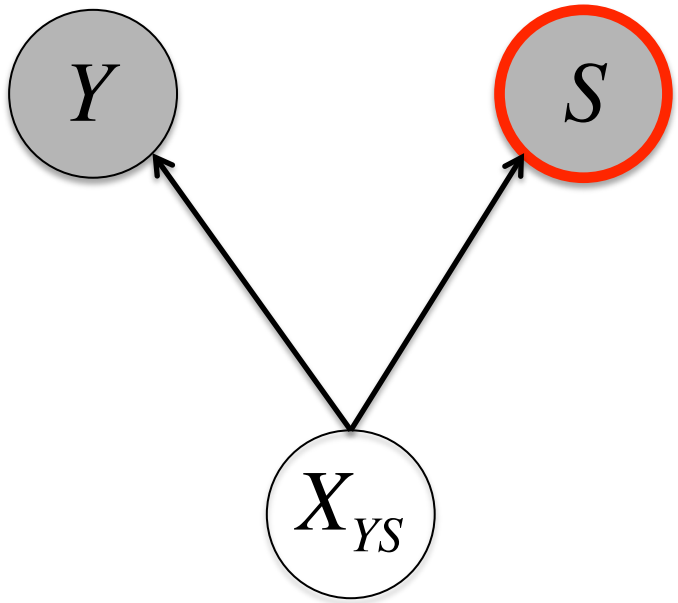
Shared Shape Space - Inference

Step 1: find a latent space point in X_{YS} which produces Y that best segments the image



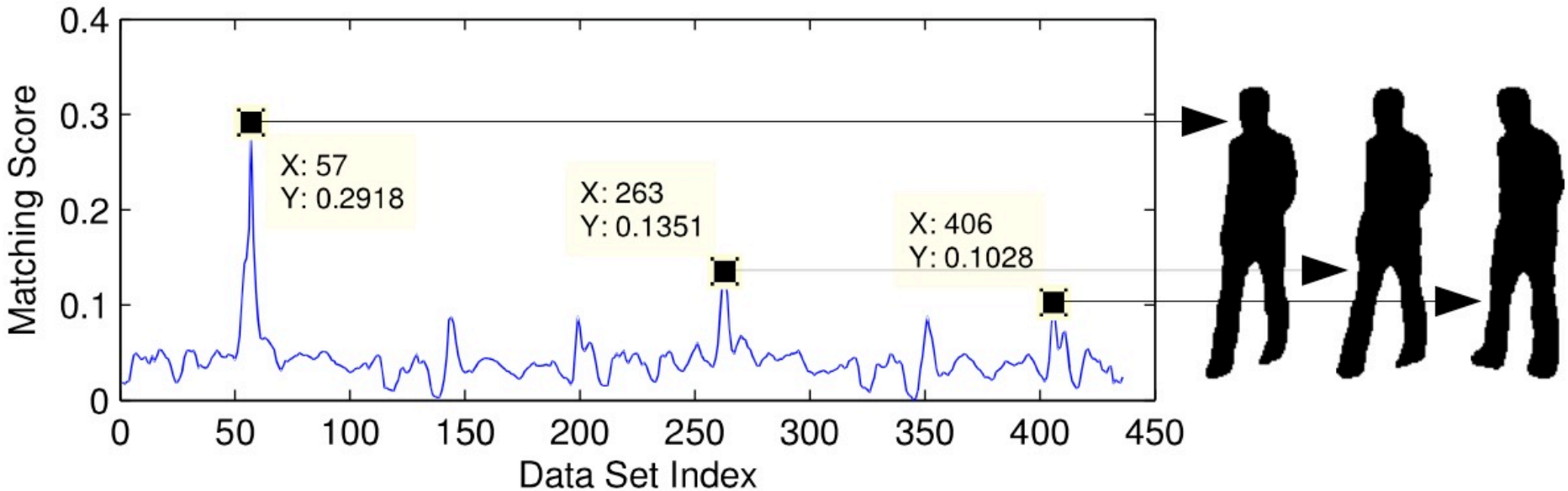
Shared Shape Space - Inference

Step 2: obtain a similarity function S



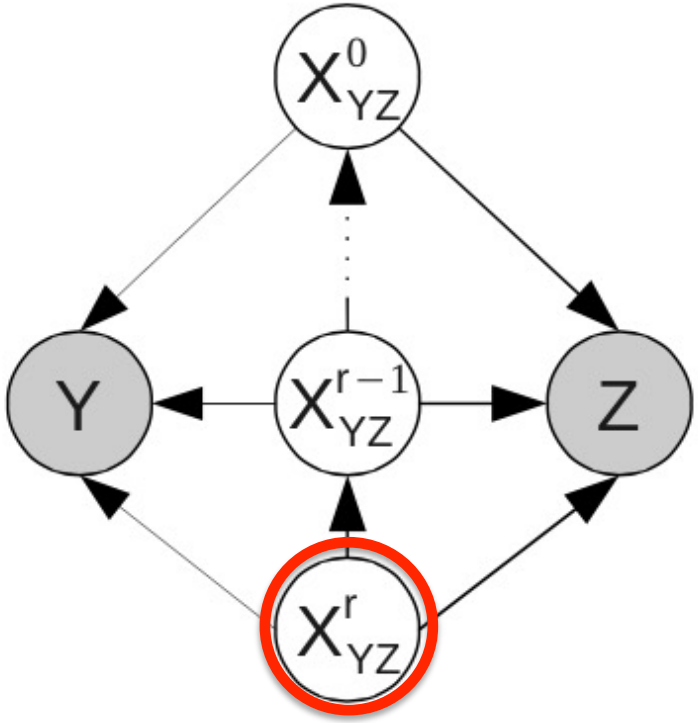
Shared Shape Space - Inference

Step 3: pick the peaks in s , which are latent points in space X_{YZ}



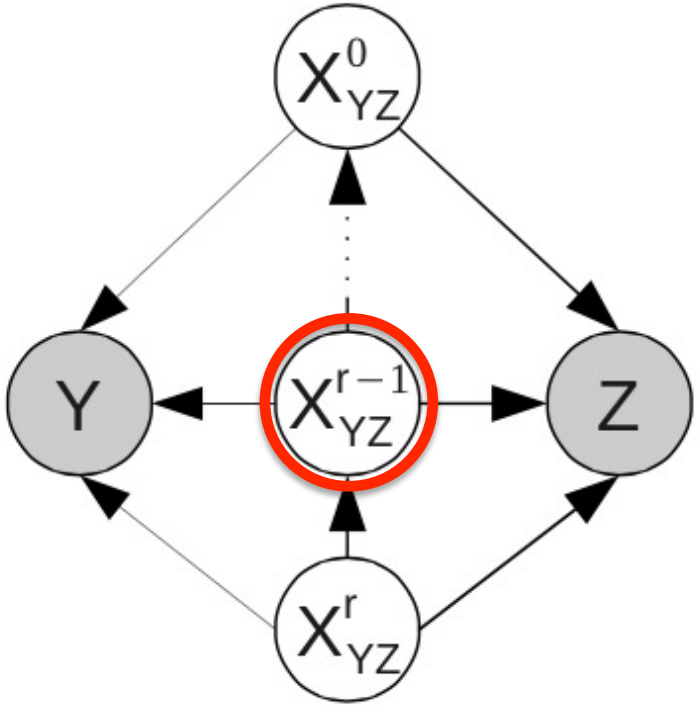
Shared Shape Space - Inference

Step 4: take these points as initializations, and search for latent points in X_{YZ} that best segments the image.



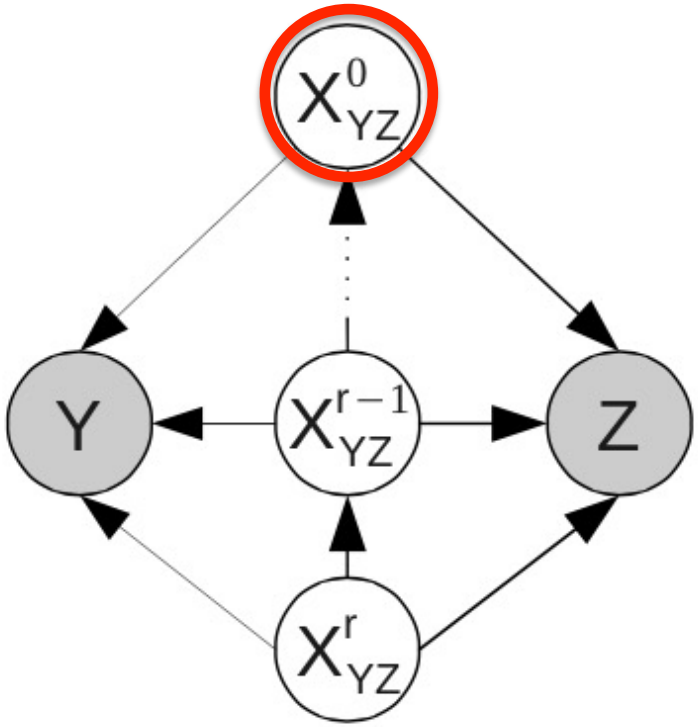
Shared Shape Space - Inference

Step 5: obtain latent points in the upper layer and continue.



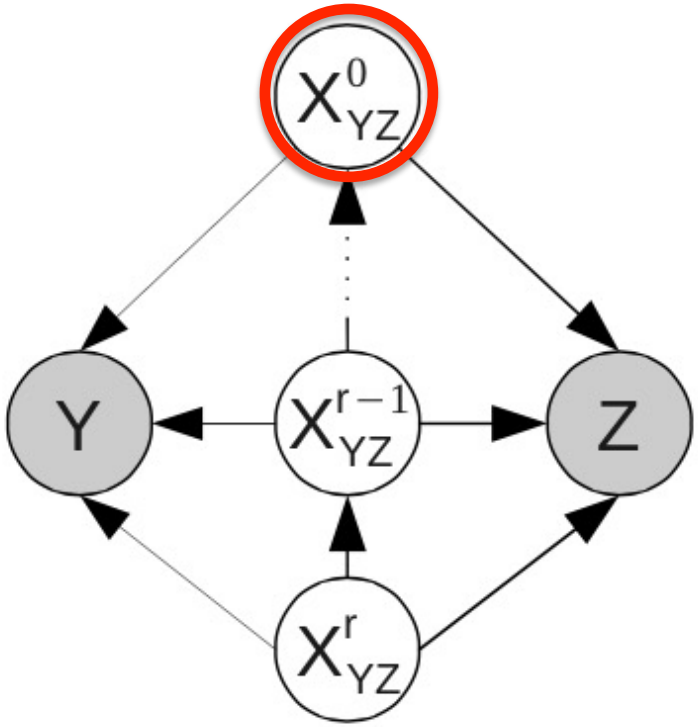
Shared Shape Space - Inference

Step 5: obtain latent points in the upper layer and continue.



Shared Shape Space - Inference

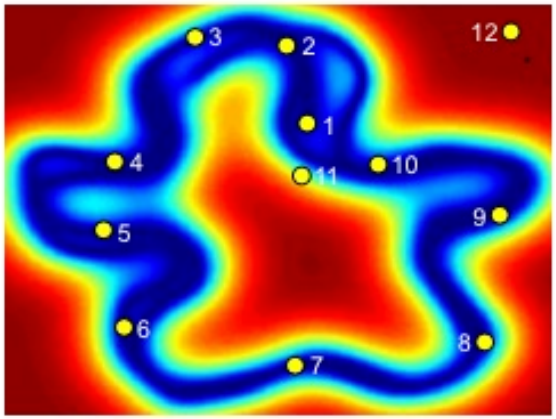
Step 5: obtain latent points in the upper layer and continue.

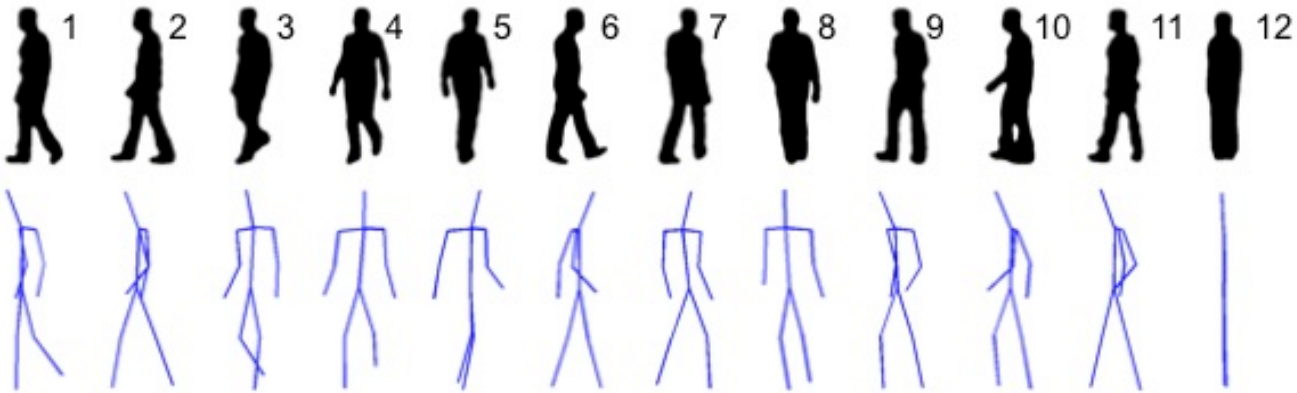


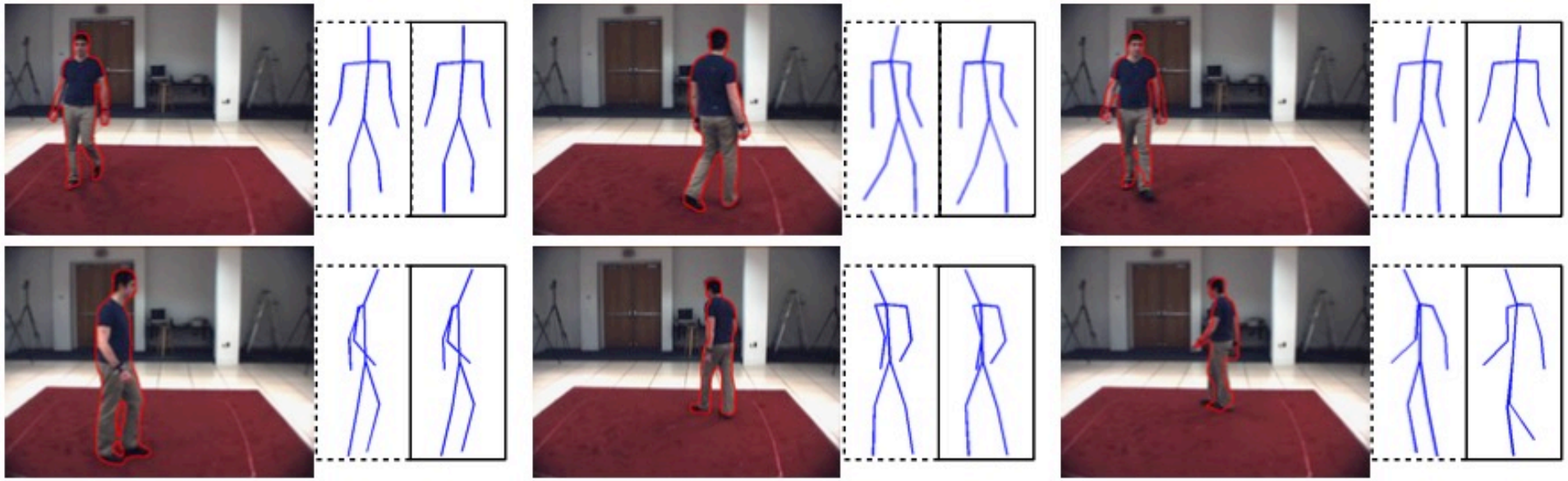
Please refer to the paper to see how to find the latent points for best segmenting the image in each step! (Also use GPLVM)

Shared Shape Space - Results

Tracking

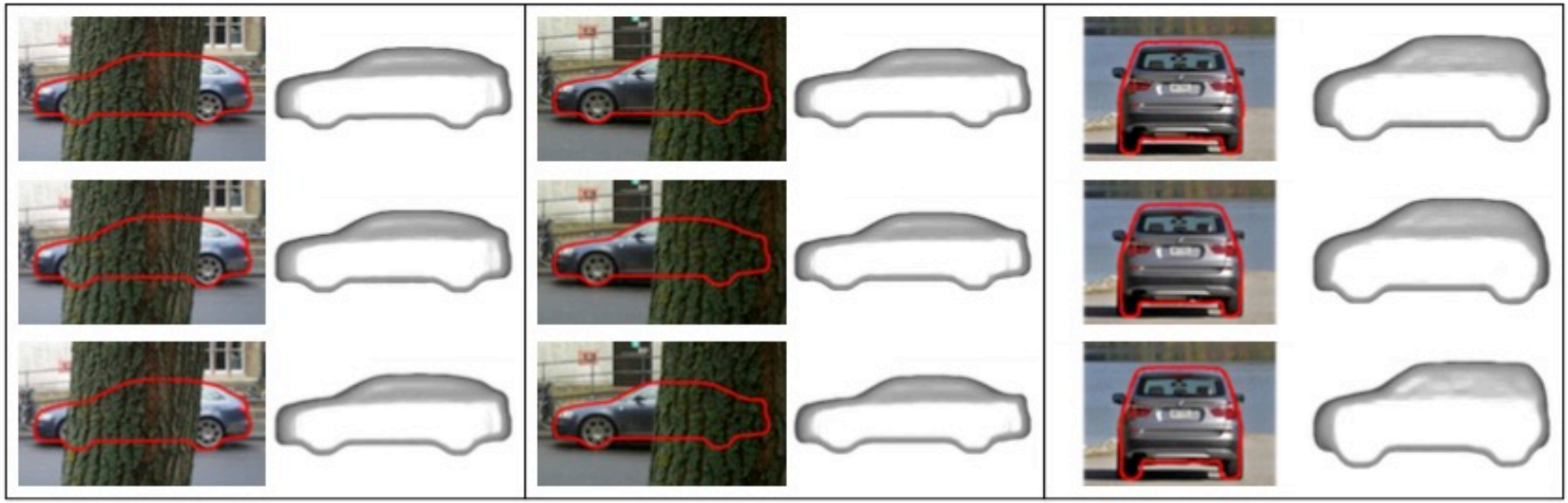






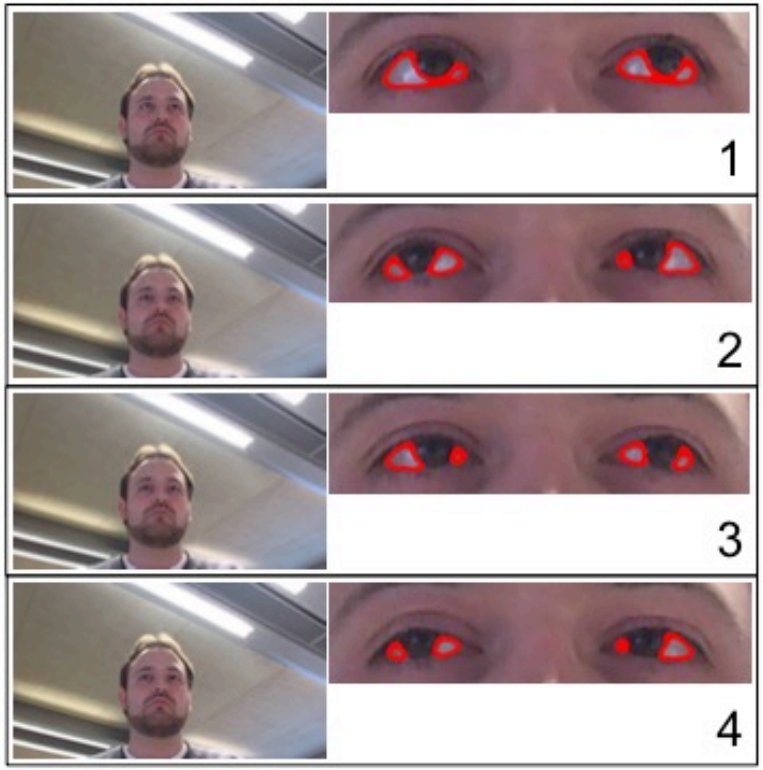
Shared Shape Space - Results

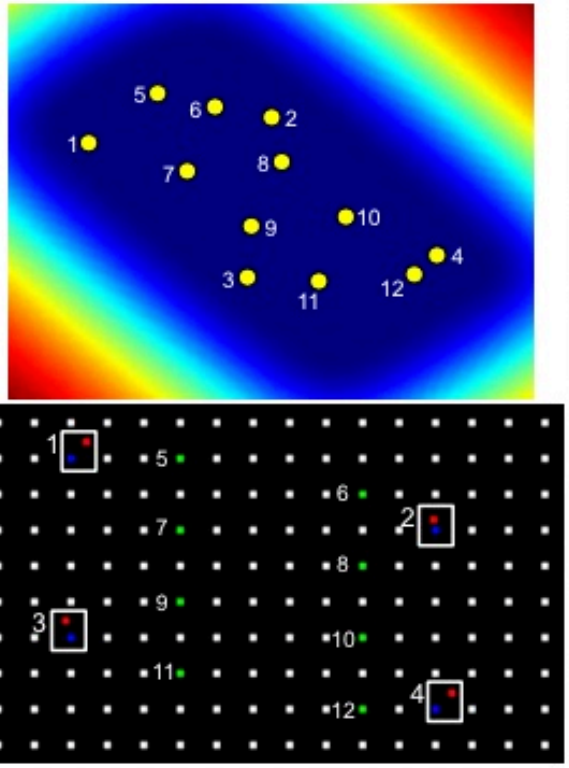
Shape from single images

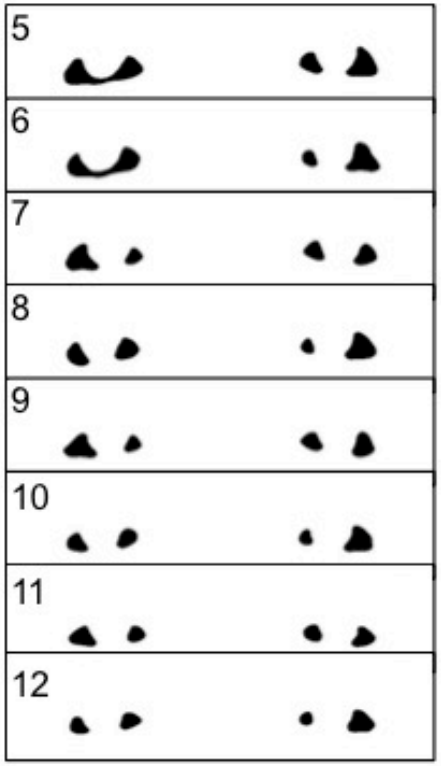


Shared Shape Space - Results

Gaze tracking







Outline

- A traditional generative model
- Shared shape space
- A benchmark for 3D object detection

A Benchmark: Beyond PASCAL

Benchmark for what?

A Benchmark: Beyond PASCAL

Advantage over previous benchmarks

- Background clutter
- Occluded/truncated objects
- Large number of classes and intra-class variation
- Large number of viewpoints

Advantages

- Background clutter
- Occluded/truncated objects
- Large number of classes and intra-class variation
- Large number of viewpoints

Advantages

- Background clutter
- Occluded/truncated objects
- Large number of classes and intra-class variation
- Large number of viewpoints

Advantages

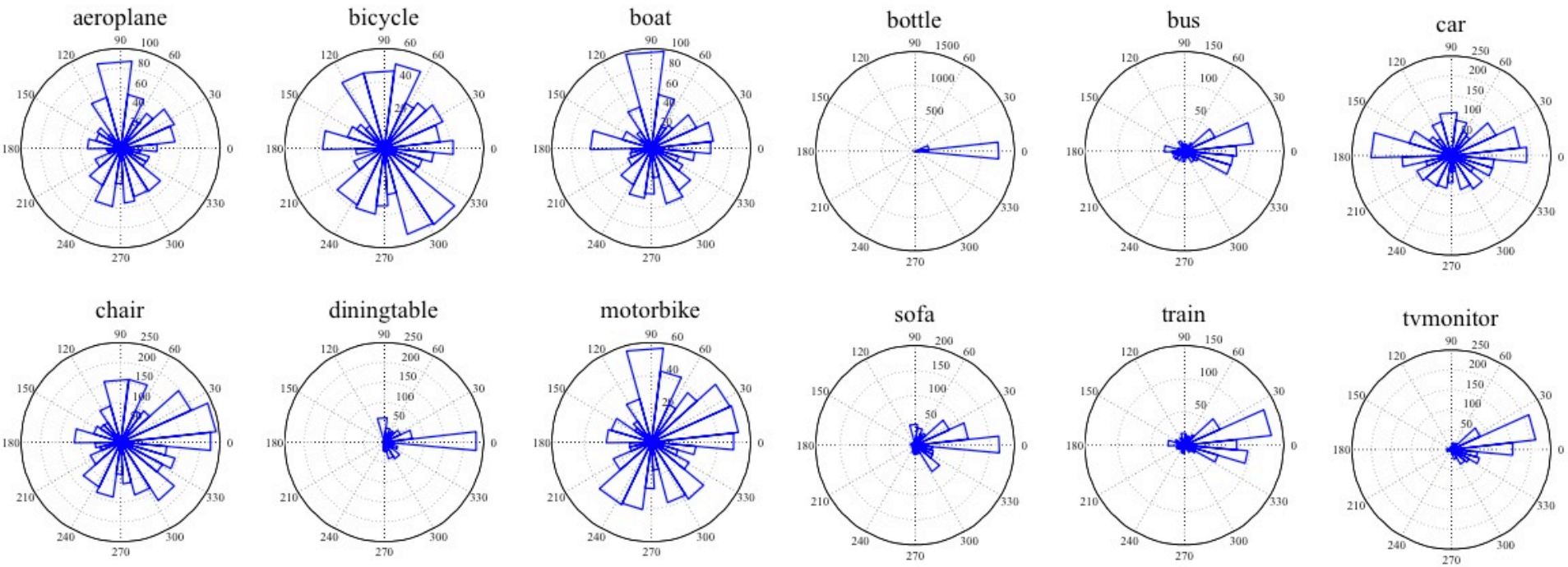
- Background clutter
- Occluded/truncated objects
- Large number of classes and intra-class variation
- Large number of viewpoints

Advantages

- Background clutter
- Occluded/truncated objects
- Large number of classes and intra-class variation
- Large number of viewpoints

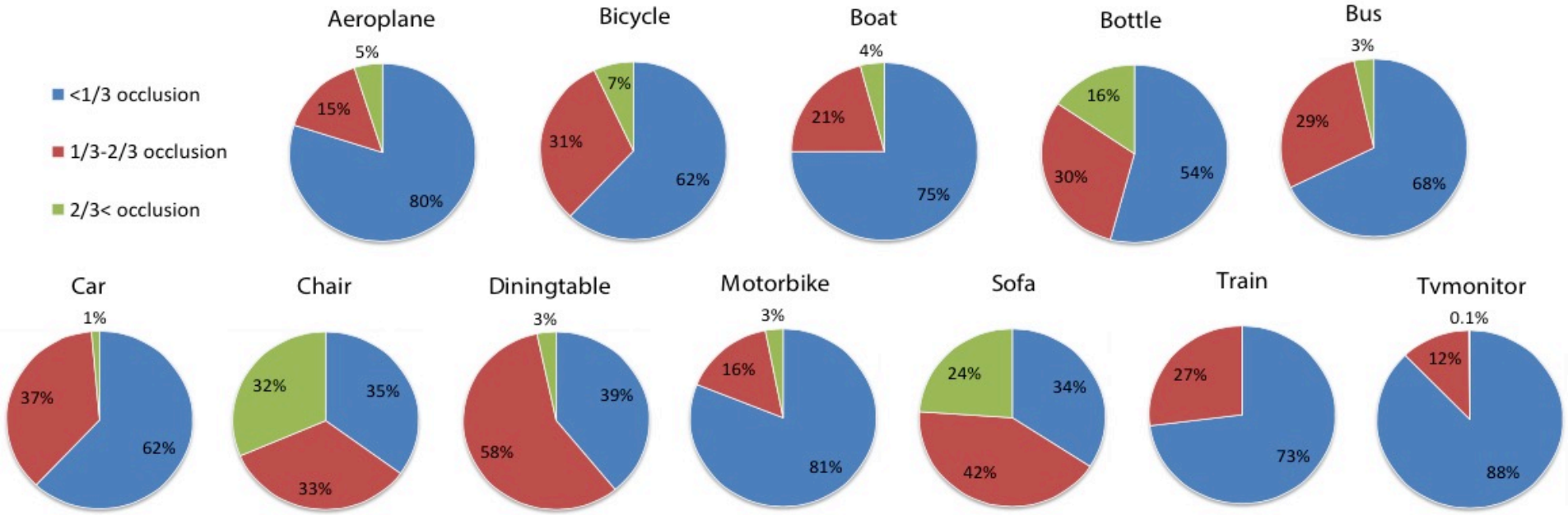
A Benchmark: Beyond PASCAL

Statistics of azimuth distribution



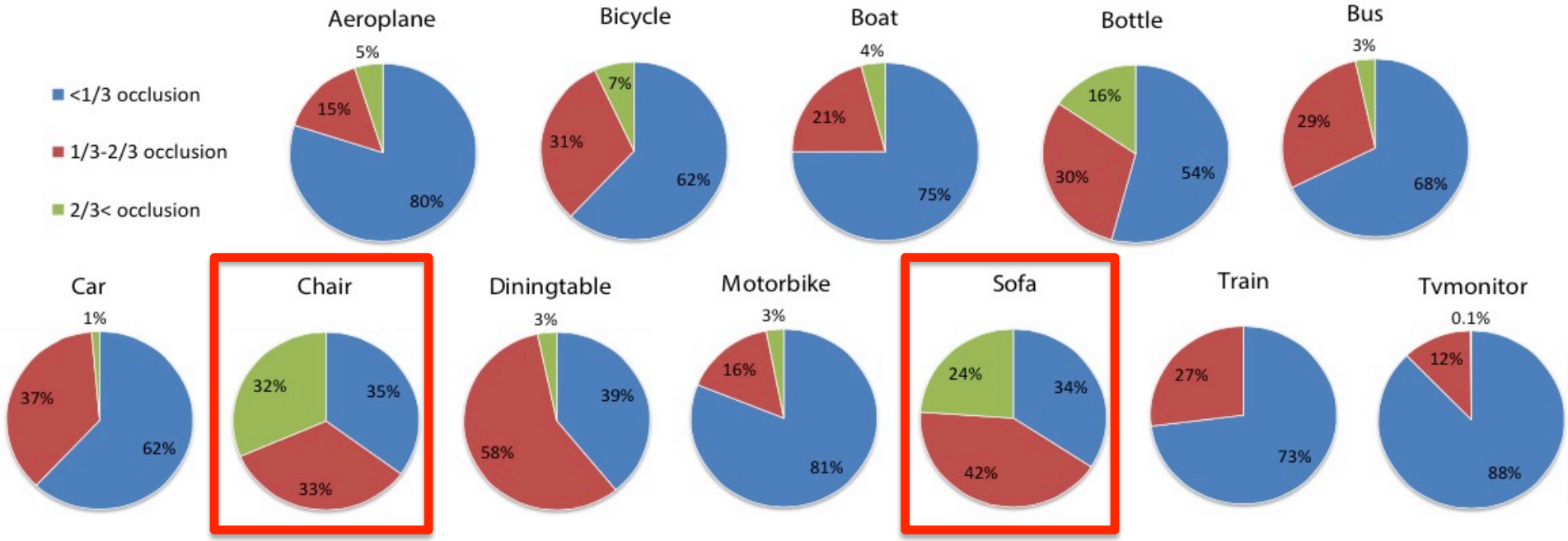
A Benchmark: Beyond PASCAL

Statistics of occlusion distribution

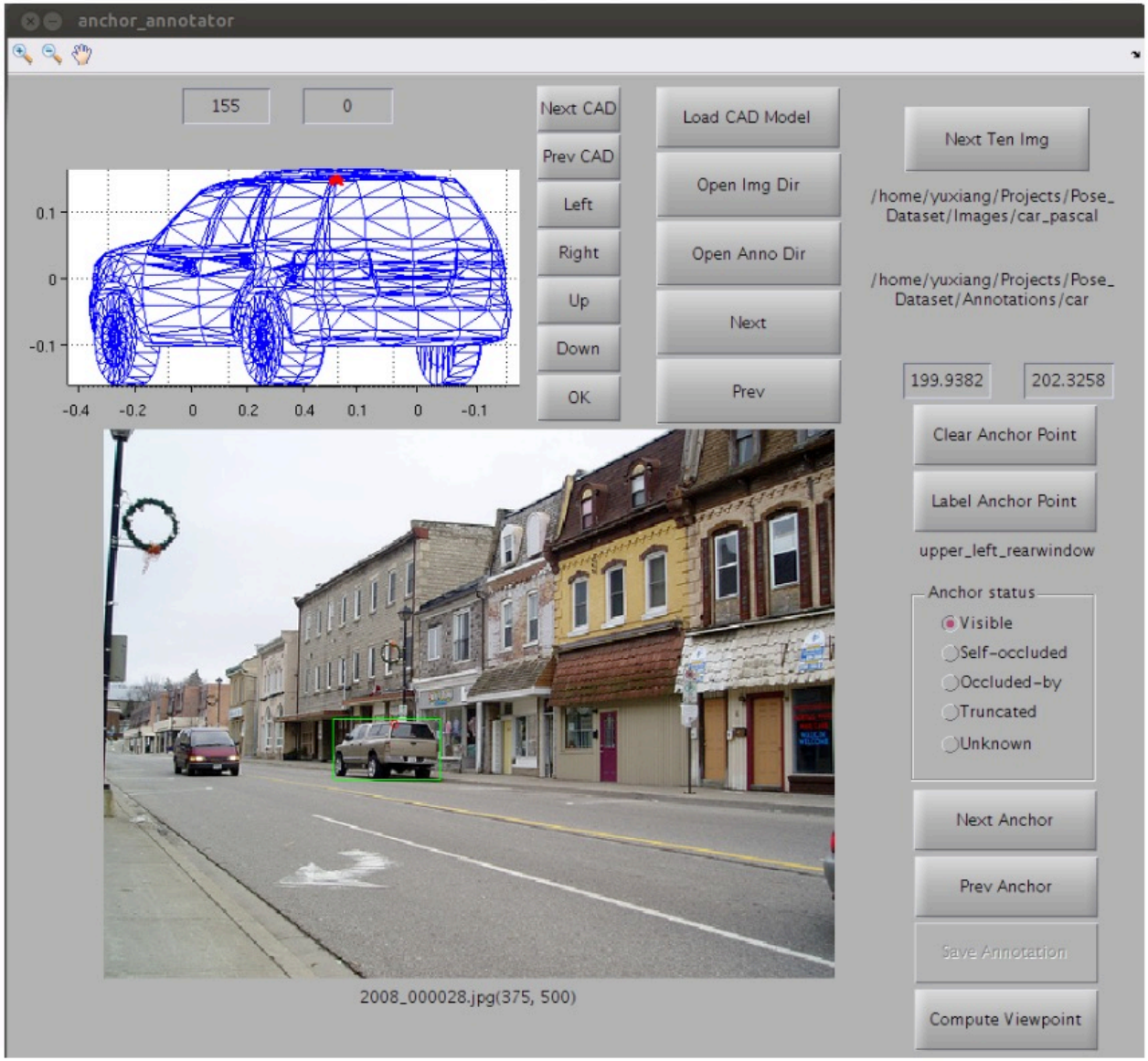


A Benchmark: Beyond PASCAL

Statistics of occlusion distribution



A Benchmark: Beyond PASCAL



Papers to be covered in this lecture

Tracking

- Simultaneous Monocular 2D Segmentation, 3D Pose Recovery and 3D Reconstruction [Prisacariu et al. 12]
- Monocular Multiview Object Tracking with 3D Aspect Parts [Xiang et al. 14 (ECCV)]

Tracking

Input: a sequence of frames

Output: segmentation, pose, 3D reconstruction per frame

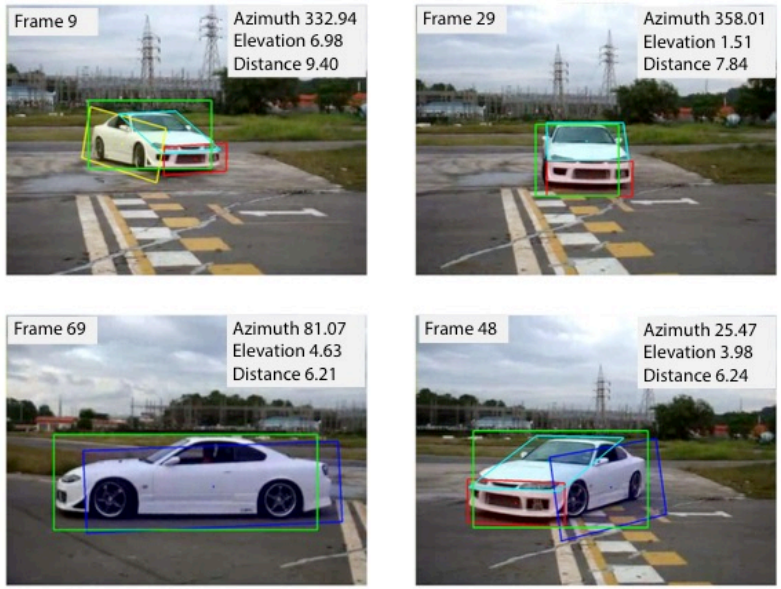
Outline

Paper	Shape representation	Motion prior	Shape consistency
Prisacariu et al. 12	DCT (same as Shared Shape Space)	None	Constrained to be the same 3D shape
Xiang et al. 14	Assembly of 3D aspect parts	Gaussian priors	Gaussian priors

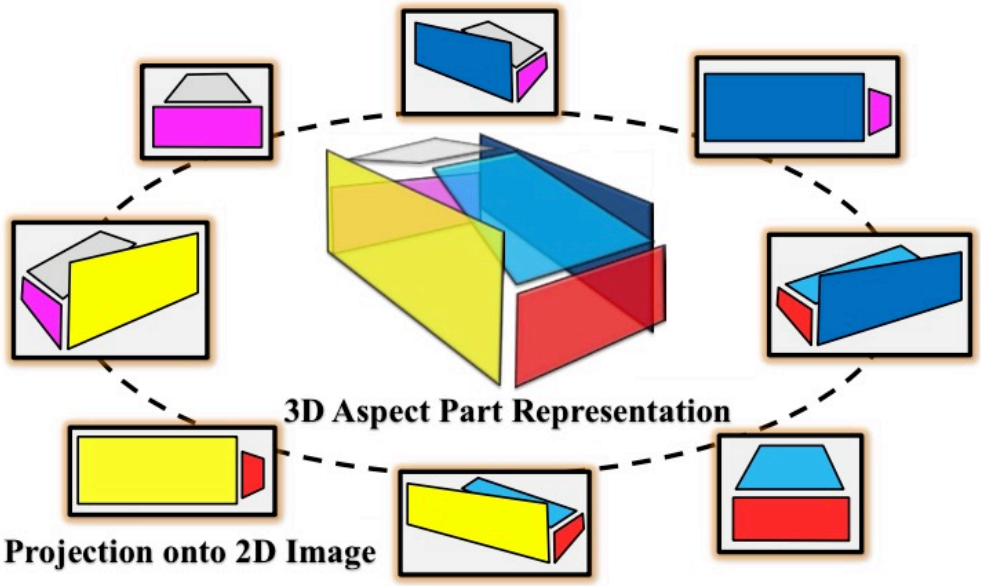
Outline

Paper	Shape representation	Motion prior	Shape consistency
Prisacariu et al. 12	DCT (same as Shared Shape Space)	None	Constrained to be the same 3D shape
Xiang et al. 14	Assembly of 3D aspect parts	Gaussian priors	Gaussian priors

Tracking with 3D Aspect Parts



Frames



3D aspect parts

Tracking with 3D Aspect Parts

To maximize the following posterior distribution:

$$P(X_t, V_t | Z_{1:t}) \propto$$

$$\underbrace{P(Z_t | X_t, V_t)}_{\text{likelihood}} \int \underbrace{P(X_t, V_t | X_{t-1}, V_{t-1})}_{\text{motion prior}} \underbrace{P(X_{t-1}, V_{t-1} | Z_{1:t-1})}_{\text{posterior at time t-1}} dX_{t-1} dV_{t-1}$$

X_t : positions of aspect parts

V_t : viewpoints

Z_t : image

Tracking with 3D Aspect Parts

To maximize the following posterior distribution:

$$P(X_t, V_t | Z_{1:t}) \propto \underbrace{\boxed{P(Z_t | X_t, V_t)}}_{\text{likelihood}} \int \underbrace{P(X_t, V_t | X_{t-1}, V_{t-1})}_{\text{motion prior}} \underbrace{P(X_{t-1}, V_{t-1} | Z_{1:t-1})}_{\text{posterior at time t-1}} dX_{t-1} dV_{t-1}$$

X_t : positions of aspect parts

V_t : viewpoints

Z_t : image

Likelihood term

Assumption of independency:

$$P(Z_t|X_t, V_t) = \prod_{i=1}^{n} P(Z_t|X_{it}, V_t)$$

X_{it} : position of an aspect part

Likelihood term

Using classifiers in both category and instance level:

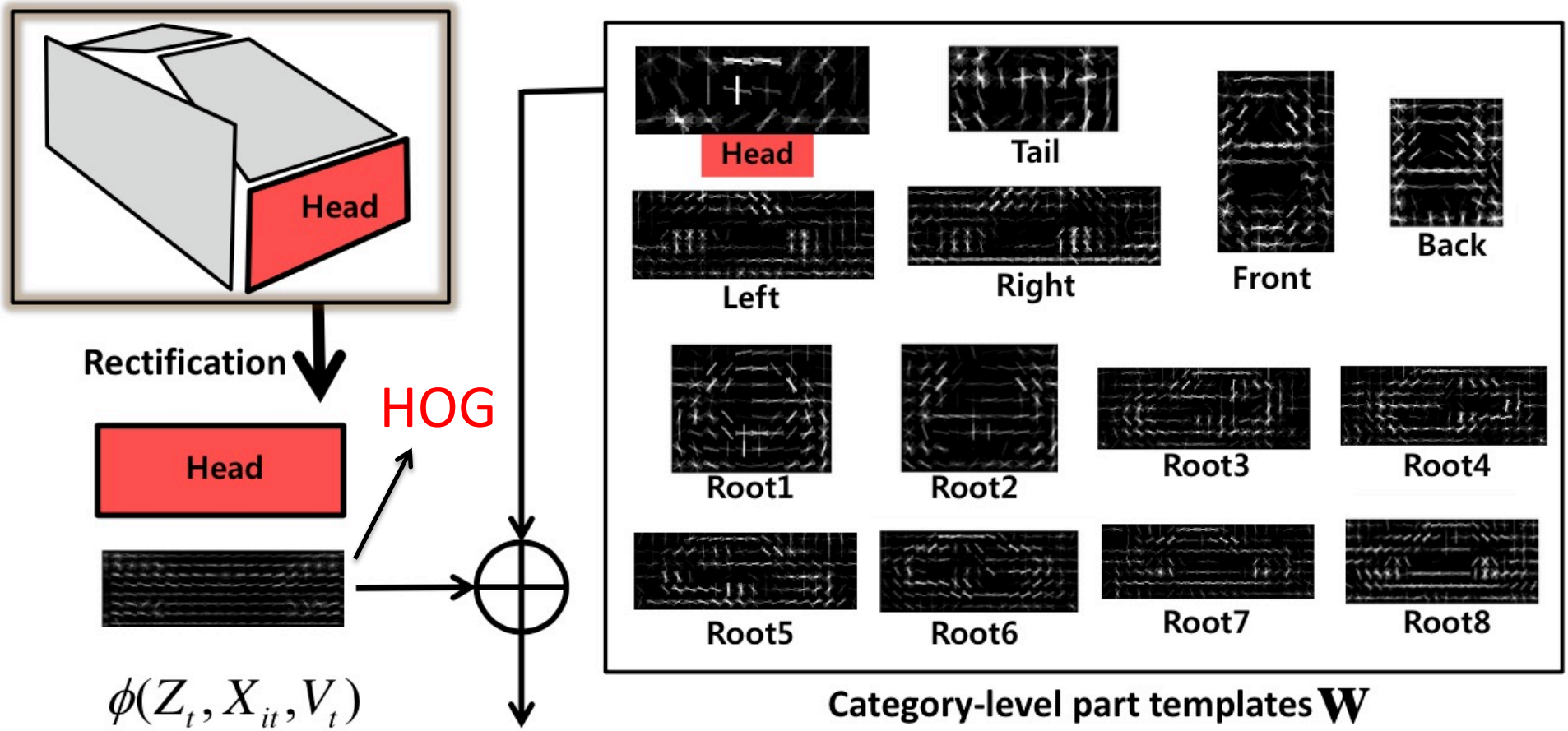
$$P(Z_t|X_{it}, V_t) \propto \exp\Big(\Lambda_{\text{category}}(Z_t, X_{it}, V_t) + \Lambda_{\text{online}}(Z_t, X_{it}, V_t)\Big)$$

$$\Lambda_{\text{category}}(Z_t, X_{it}, V_t) = \begin{cases} \mathbf{w}_i^T \phi(Z_t, X_{it}, V_t), & \text{if visible} \\ \alpha_i, & \text{if self-occluded,} \end{cases}$$

$$\Lambda_{\text{online}}(Z_t, X_{it}, V_t) = \begin{cases} \mathbf{H}_i(\psi(Z_t, X_{it}, V_t)), & \text{if visible} \\ \lambda_0, & \text{if self-occluded,} \end{cases}$$

Likelihood term

Descriptor of a rectified aspect part $\phi(Z_t, X_{it}, V_t)$



Extracting Haar-like features for instance-level templates

Tracking with 3D Aspect Parts

To maximize the following posterior distribution:

$$P(X_t, V_t | Z_{1:t}) \propto$$

$$\underbrace{P(Z_t | X_t, V_t)}_{\text{likelihood}} \int \underbrace{P(X_t, V_t | X_{t-1}, V_{t-1})}_{\text{motion prior}} \underbrace{P(X_{t-1}, V_{t-1} | Z_{1:t-1})}_{\text{posterior at time t-1}} dX_{t-1} dV_{t-1}$$

X_t : positions of aspect parts

V_t : viewpoints

Z_t : image

Tracking with 3D Aspect Parts

To maximize the following posterior distribution:

$$P(X_t, V_t | Z_{1:t}) \propto$$

$$\underbrace{P(Z_t | X_t, V_t)}_{\text{likelihood}} \int \underbrace{\boxed{P(X_t, V_t | X_{t-1}, V_{t-1})}}_{\text{motion prior}} \underbrace{P(X_{t-1}, V_{t-1} | Z_{1:t-1})}_{\text{posterior at time t-1}} dX_{t-1} dV_{t-1}$$

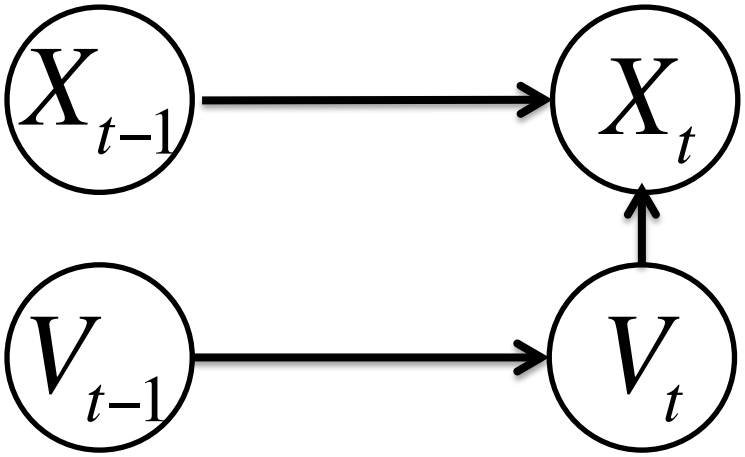
X_t : positions of aspect parts

V_t : viewpoints

Z_t : image

Motion prior term

$$
\begin{aligned}
& P(X_t, V_t | X_{t-1}, V_{t-1}) \\
&= P(X_t | X_{t-1}, V_{t-1}, V_t) P(V_t | X_{t-1}, V_{t-1}) \\
&= P(X_t | X_{t-1}, V_t) P(V_t | V_{t-1}),
\end{aligned}
$$



Motion prior term

Assumption of independency:

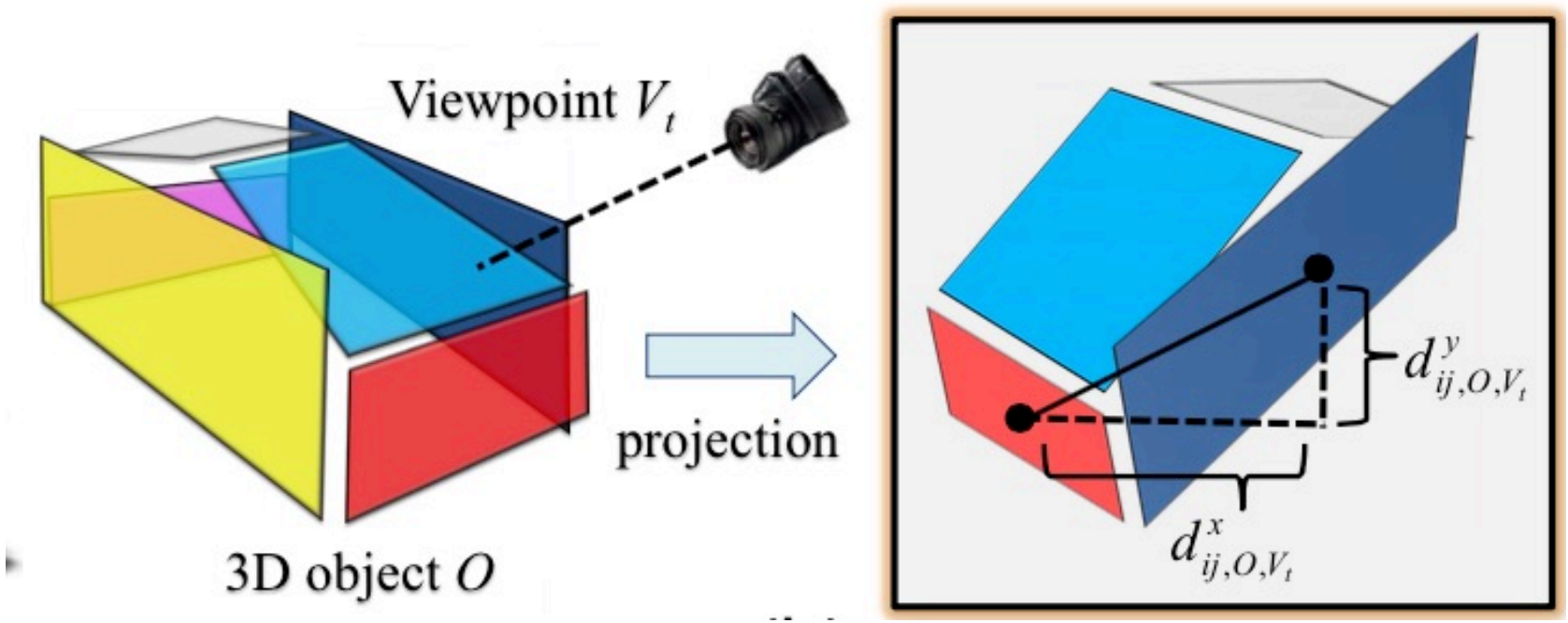
$$P(X_t|X_{t-1}, V_t) \propto \prod_{i=1}^{n} P(X_{it}|X_{i(t-1)}) \prod_{(i,j)} \Lambda(X_{it}, X_{jt}, V_t),$$

$$P(X_{it}|X_{i(t-1)}) \sim \mathcal{N}(X_{i(t-1)}, \sigma_x^2, \sigma_y^2)$$

Motion prior term

Pairwise term:

$$\Lambda(X_{it}, X_{jt}, V_t) = P(\Delta_t(x_i, x_j)|V_t)P(\Delta_t(y_i, y_i)|V_t),$$
$$P(\Delta_t(x_i, x_j)|V_t) \sim \mathcal{N}(d^x_{ij,O,V_t}, \sigma^2_{dx}),$$
$$P(\Delta_t(y_i, y_j)|V_t) \sim \mathcal{N}(d^y_{ij,O,V_t}, \sigma^2_{dy}),$$



Inference

MCMC sampling

input : A video sequence $Z_{1:T}$, initial 3D aspect parts and viewpoint (X_1, V_1)
output: 3D aspect parts and viewpoints for the target in the video $(X_t, V_t)_{t=1}^{T}$

Initialize samples $(X_1^{(r)}, V_1^{(r)})_{r=1}^{N}$ *for the first frame by sampling viewpoints and part locations according to the motion prior* (6) *based on* (X_1, V_1);
for $t \leftarrow 2$ **to** T **do**
  Initialize the MCMC sampler: randomly select a sample $(X_{t-1}^{(r)}, V_{t-1}^{(r)})$ *as the initial state of the* (X_t, V_t) *Markov chain*;
  repeat
    Sample a new viewpoint from the Gaussian proposal density $Q(V_t'; V_t)$;
    Compute the visibility of 3D aspect parts under viewpoint V_t';
    foreach *part* i *visible in both* V_t' *and* V_t **do**
      Sample its location from the Gaussian proposal density $Q(X_{it}'; X_{it})$;
    end
    foreach *part* i *visible in* V_t' *but not in* V_t **do**
      Compute its location X_{it}' *using the mean distance with respect to other visible parts according to the pairwise distributions* (10);
    end
    Compute the acceptance ratio

$$a = \min\left(1, \frac{P(X_t', V_t' | Z_{1:t}) Q(X_t; X_t') Q(V_t; V_t')}{P(X_t, V_t | Z_{1:t}) Q(X_t'; X_t) Q(V_t'; V_t)}\right); \tag{12}$$

    Accept the sample (X_t', V_t') *with probability* a. *If accepted,* $(X_t, V_t) \leftarrow (X_t', V_t')$. *Otherwise, leave* (X_t, V_t) *unchanged*;
  until N *samples are accepted*;
  Obtain the new sample set $(X_t^{(r)}, V_t^{(r)})_{r=1}^{N}$, *and find the MAP among it as the tracking output for frame* t;
end

Results - Tracking

Video	MIL [2]	L1 [3]	TLD [20]	Struct [15]	DPM [12]+PF	Category Model	Full Model
Race1	0.34	0.39	0.20	0.36	0.68	0.68	**0.69**
Race2	0.49	0.49	0.28	0.50	**0.74**	**0.74**	0.73
Race3	0.36	0.26	0.25	0.44	0.74	0.74	**0.77**
Race4	0.53	0.56	0.47	0.63	**0.76**	**0.76**	**0.76**
Race5	0.29	0.54	0.28	0.26	0.63	0.63	**0.68**
Race6	0.27	0.53	0.48	0.29	0.76	0.76	**0.77**
SUV1	0.58	**0.81**	0.56	0.60	0.78	0.78	0.78
SUV2	0.18	0.12	0.53	0.24	**0.77**	**0.77**	**0.77**
Sedan	0.26	0.23	0.33	0.30	**0.78**	**0.78**	**0.78**
Mean	0.37	0.44	0.38	0.40	0.74	0.74	**0.75**
KITTI01	0.20	0.40	0.44	0.33	0.65	0.64	**0.69**
KITTI02	0.28	0.18	0.20	0.12	0.26	0.26	**0.32**
KITTI03	0.37	**0.59**	0.42	0.36	0.20	0.19	0.50
KITTI04	0.31	0.12	**0.36**	0.34	0.67	0.33	0.33
KITTI05	0.40	0.32	0.51	0.41	0.54	**0.73**	0.72
KITTI06	0.64	0.21	0.54	**0.65**	**0.65**	**0.65**	0.56
KITTI07	0.12	0.33	0.03	0.28	**0.66**	0.65	**0.66**
KITTI08	0.58	0.13	0	0.66	**0.74**	**0.74**	0.72
KITTI09	0.18	0.15	0	0.17	0.18	0.51	**0.52**
KITTI10	0.33	0.46	0.41	0.35	**0.68**	**0.68**	**0.68**
KITTI11	0.28	0.23	0.24	0.28	**0.71**	**0.71**	0.68
Mean	0.34	0.28	0.29	0.36	0.54	0.55	**0.58**
06_car [20]	0.19	0.52	**0.85**	0.48	0.70	0.67	0.70

Results – Viewpoint and 3D recovery

	Viewpoint Estimation			3D Aspect Part Localization		
Video	Full Model	Category Model	ALM [43]	Full Model	Category Model	ALM [43]
Race1	**0.67/18.73°**	0.59/22.88°	0.52/42.62°	**0.40**	0.39	0.35
Race2	**0.77/10.83°**	0.60/12.65°	0.53/44.30°	**0.45**	0.38	0.34
Race3	**0.83**/9.28°	**0.83/7.79°**	0.64/46.08°	0.45	**0.48**	0.31
Race4	0.69/15.83°	0.68/14.67°	**0.79/13.37°**	**0.48**	0.47	0.42
Race5	0.71/**10.75°**	**0.74**/11.78°	0.54/57.79°	**0.44**	0.42	0.28
Race6	**0.43/18.47°**	0.40/21.34°	0.31/37.08°	**0.35**	**0.35**	0.29
SUV1	**0.82/7.81°**	0.75/8.52°	0.47/78.38°	**0.42**	0.40	0.24
SUV2	**0.57/19.56°**	0.45/56.33°	0.39/63.41°	**0.30**	0.23	0.18
Sedan	0.76/9.87°	0.78/**9.50°**	**0.79**/20.84°	0.44	**0.45**	0.43
Mean	**0.69/13.46°**	0.65/18.38°	0.54/47.24°	**0.41**	0.40	0.30
KITTI01	**0.95/6.54°**	0.74/8.53°	0.57/44.46°	**0.49**	0.41	0.37
KITTI02	**1.00/5.40°**	0.20/30.06°	0.33/119.54°	**0.60**	0.15	0.13
KITTI03	**0.42**/15.64°	**0.42/15.14°**	0.50/15.99°	**0.33**	**0.33**	0.24
KITTI04	0.22/27.05°	**0.25/26.03°**	0.17/58.42°	**0.22**	**0.22**	0.14
KITTI05	0.36/23.59°	0.40/**22.17°**	**0.64**/23.65°	0.23	**0.25**	**0.25**
KITTI06	0.31/21.63°	0.29/21.58°	**0.59/20.29°**	0.21	0.21	**0.23**
KITTI07	**0.96/6.86°**	0.89/7.92°	0.70/24.50°	**0.48**	**0.48**	0.39
KITTI08	0.57/**15.61°**	0.48/23.84°	**0.67**/23.26°	**0.37**	0.29	0.26
KITTI09	**0.50**/21.63°	0.42/78.67°	**0.50/17.60°**	**0.28**	0.16	0.23
KITTI10	**0.81/7.99°**	0.79/9.44°	0.44/56.78°	**0.39**	**0.39**	0.21
KITTI11	**0.88/9.33°**	0.78/11.80°	0.68/12.29°	0.39	0.40	**0.41**
Mean	**0.63/14.66°**	0.51/23.20°	0.53/37.89°	**0.36**	0.30	0.26

Papers to be covered in this lecture

Shape synthesis

- A Probabilistic Model for Component-based Shape Synthesis [Kalogerakis et al. 12]
- Co-constrained Handles for Deformation in Shape Collections [Yumer et al. 14]

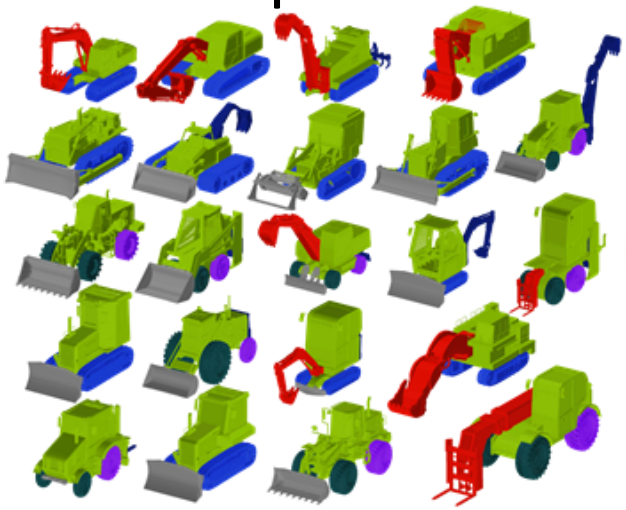
Papers to be covered in this lecture

Shape synthesis

- A Probabilistic Model for Component-based Shape Synthesis [Kalogerakis et al. 12]
- Co-constrained Handles for Deformation in Shape Collections [Yumer et al. 14]

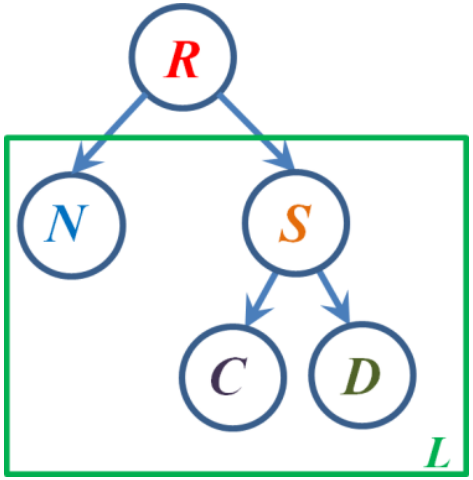
Probabilistic Model for Shape Synthesis

Input



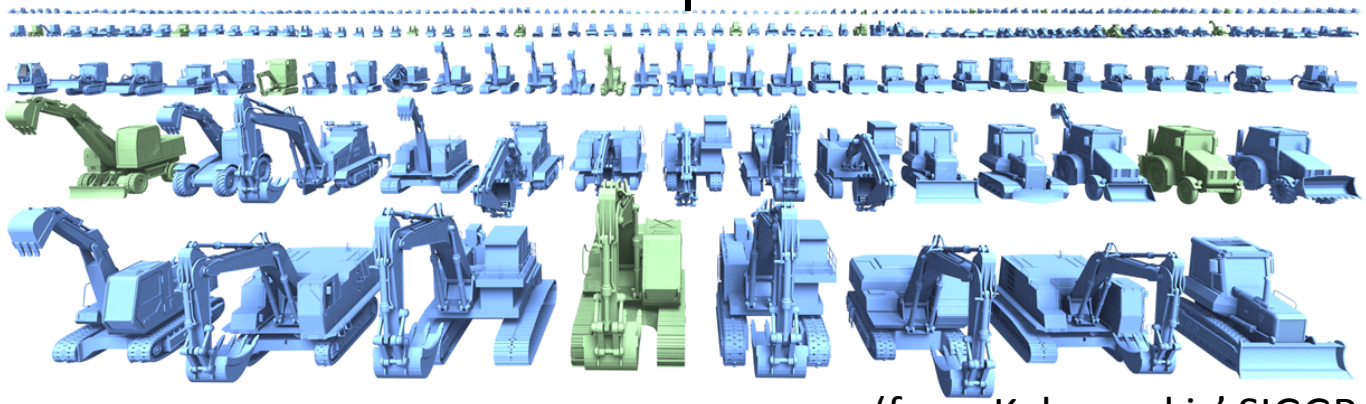
Learning stage

Probabilistic model



Synthesis stage

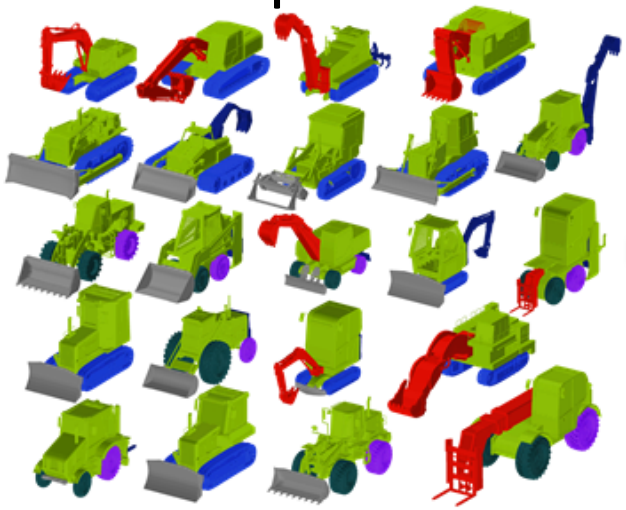
Output



(from Kalogerakis' SIGGRAPH slides)

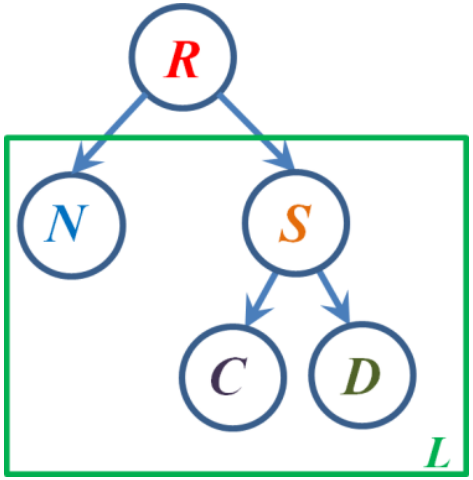
Learning stage

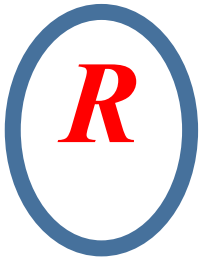
Input



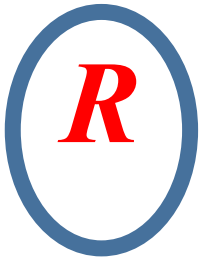
Learning stage

Probabilistic model

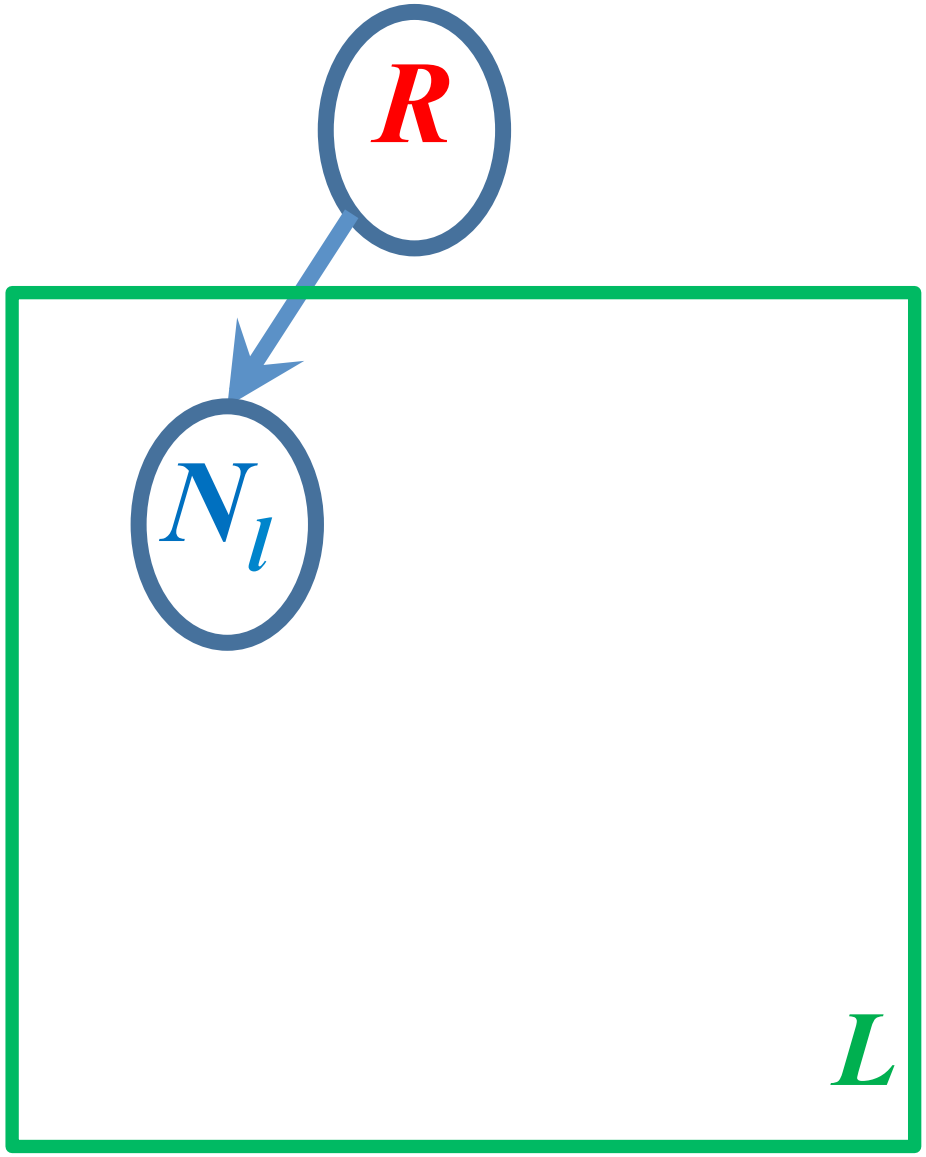




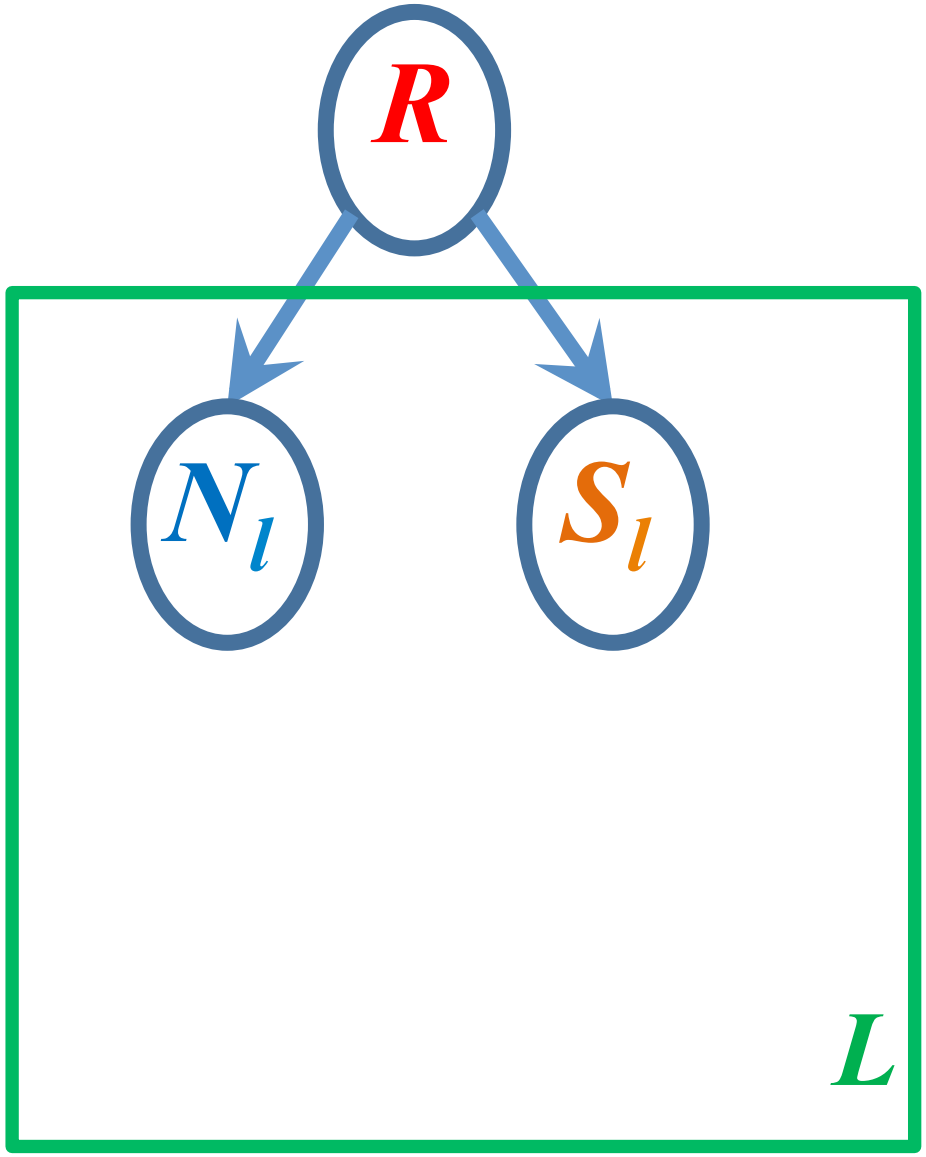
P(R)



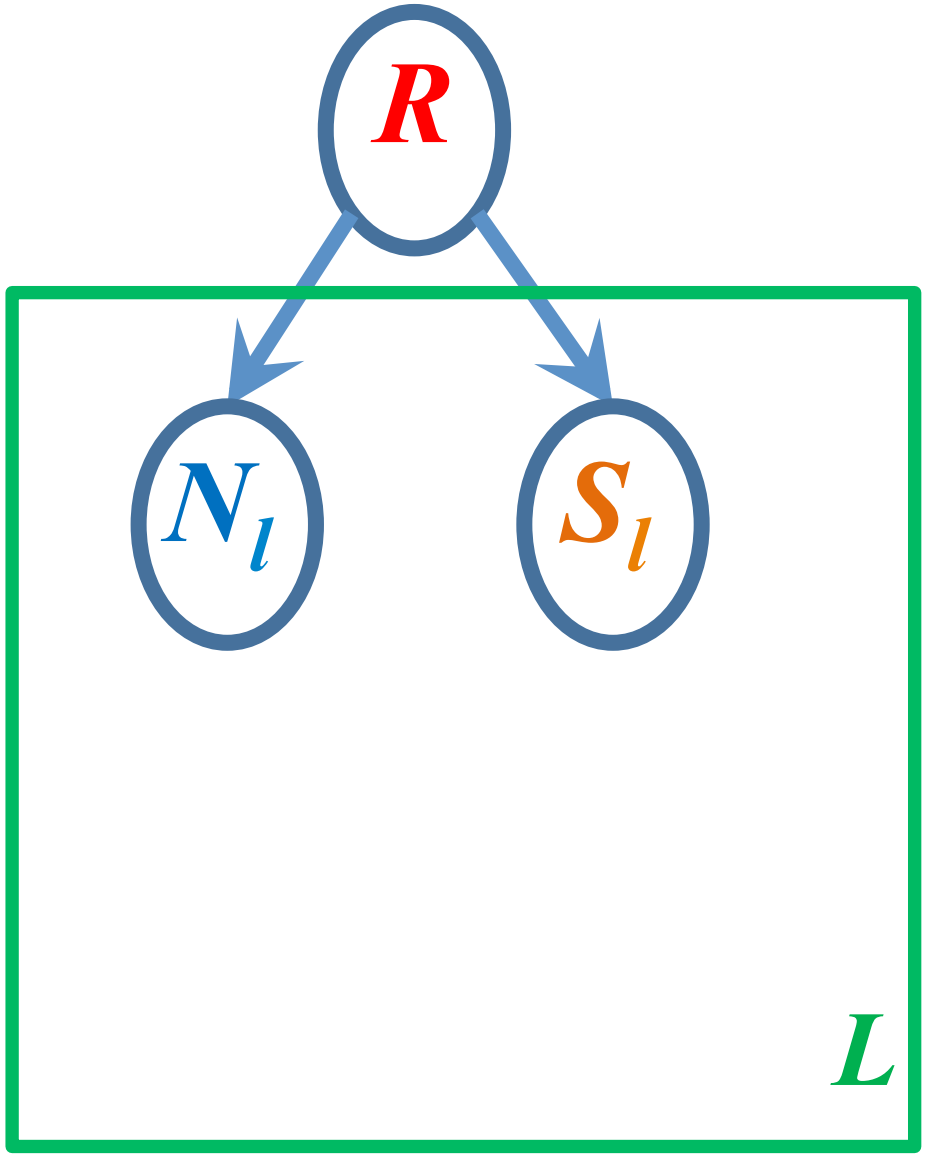
P(R)

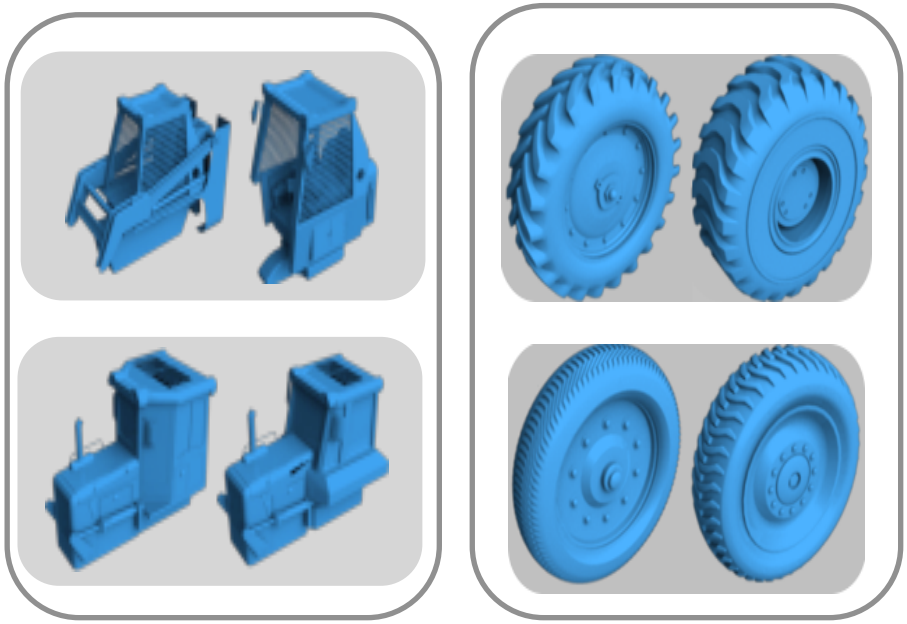


$$P(R) \prod_{l \in L} \left[P(N_l \mid R) \right]$$

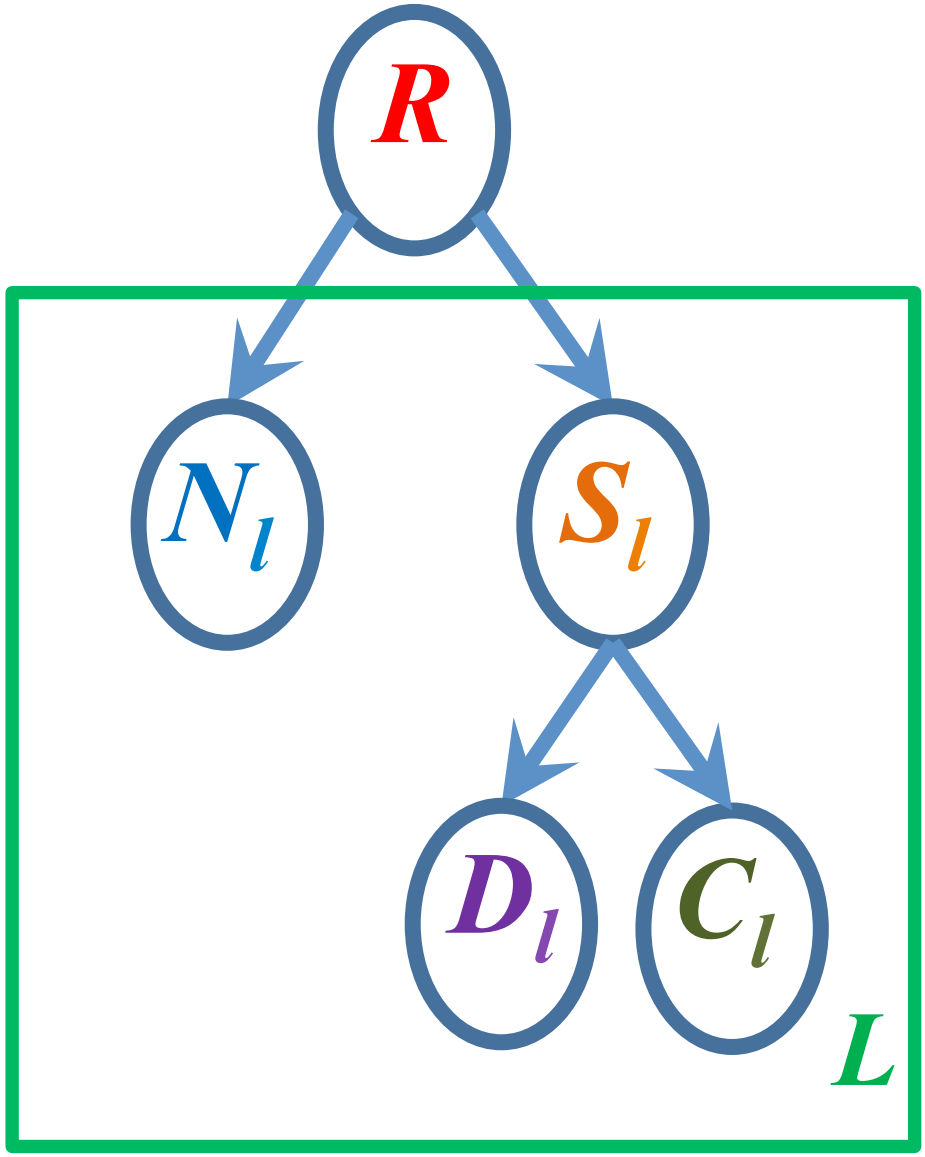


$$P(R) \prod_{l \in L} \left[P(N_l \mid R)\, P(S_l \mid R) \right]$$

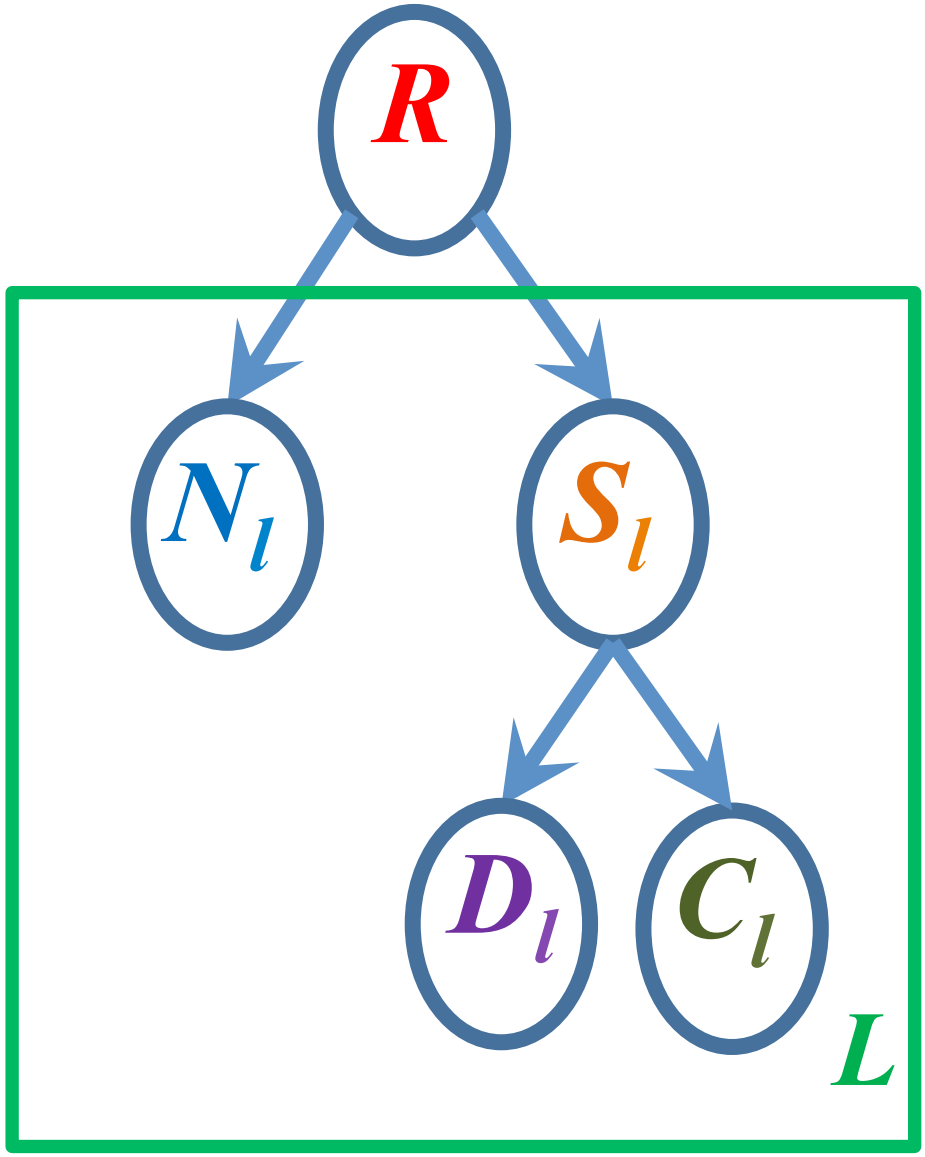


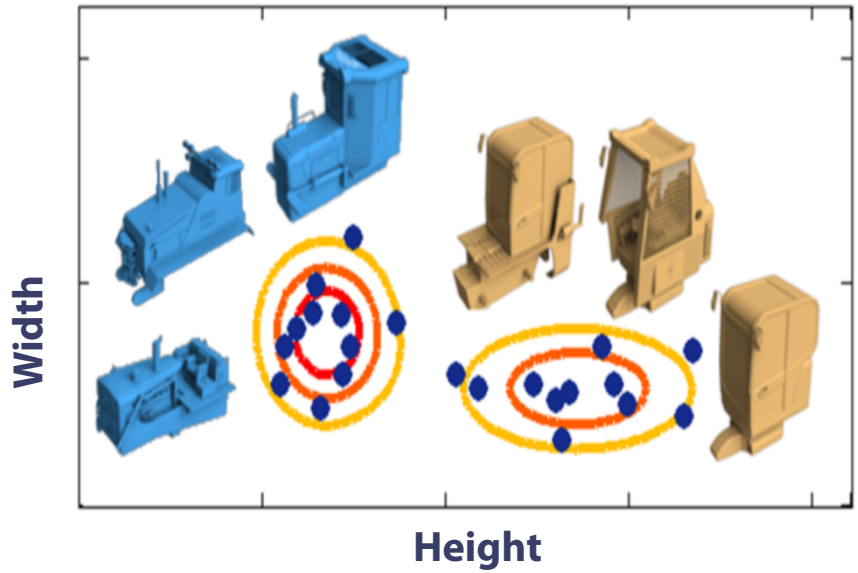


$$P(R) \prod_{l \in L} \left[P(N_l \mid R)\, P(S_l \mid R) \right]$$

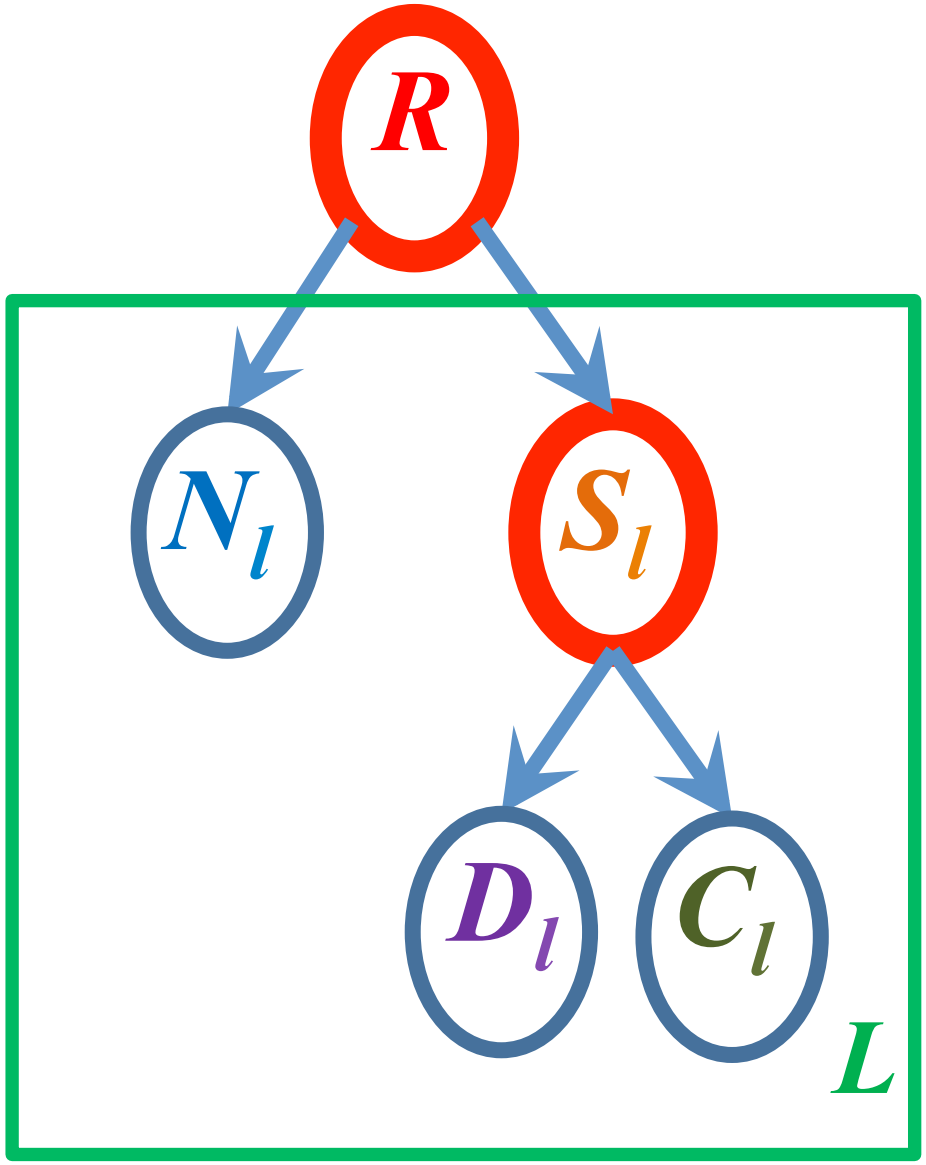


$$P(R) \prod_{l \in L} \left[P(N_l \mid R)\, P(S_l \mid R)\, P(D_l \mid S_l)\, P(C_l \mid S_l) \right]$$





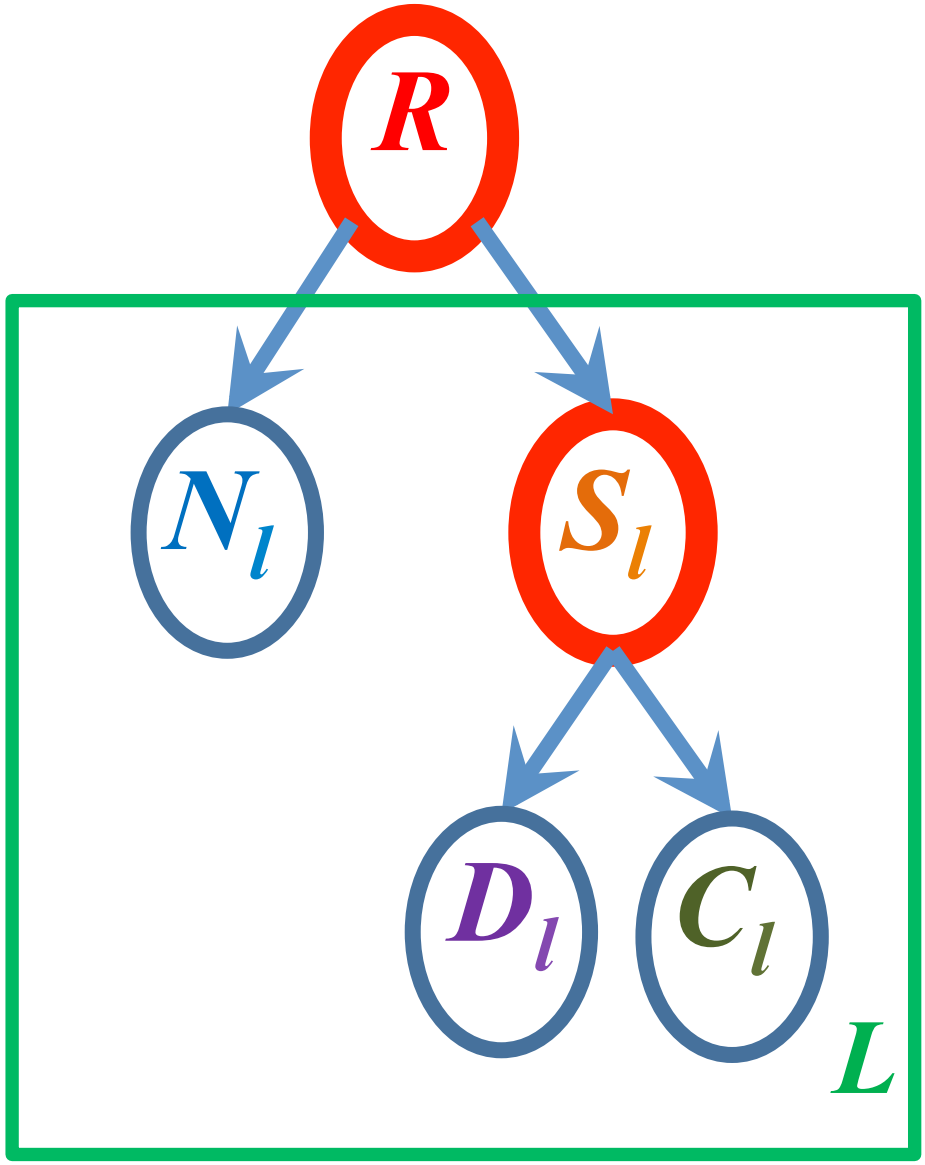
$$P(R) \prod_{l \in L} \left[P(N_l \mid R)\, P(S_l \mid R)\, P(D_l \mid S_l)\, P(C_l \mid S_l) \right]$$

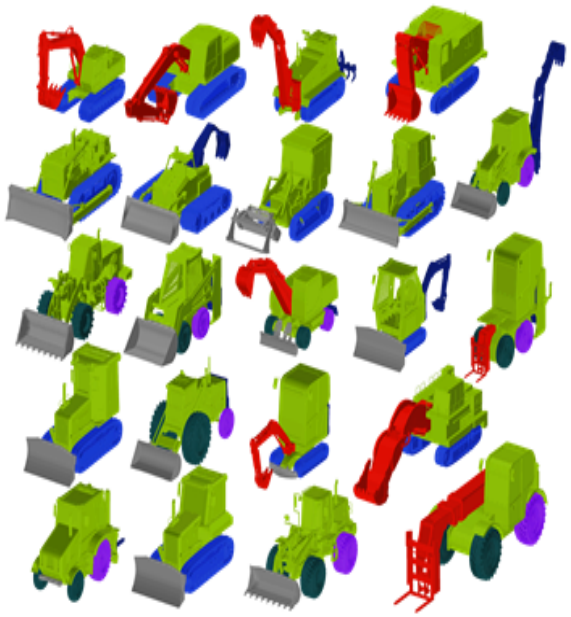


Latent object style

Latent component style

$$P(R) \prod_{l \in L} \left[P(N_l \mid R)\, P(S_l \mid R)\, P(D_l \mid S_l)\, P(C_l \mid S_l) \right]$$





Learn from training data:

latent styles

lateral edges

parameters of CPDs

Learning stage

Given observed data $\mathbf{O}$, find structure $\mathbf{G}$ that maximizes:

$$P(G \mid \mathbf{O}) = \frac{P(\mathbf{O} \mid G)P(G)}{P(\mathbf{O})}$$

Learning stage

Given observed data $\mathbf{O}$, find structure $\mathbf{G}$ that maximizes:

$$P(G \mid \mathbf{O}) = \frac{P(\mathbf{O} \mid G)P(G)}{P(\mathbf{O})}$$

Assuming uniform prior over structures, maximize **marginal likelihood**:

$$P(\mathbf{O} \mid G) = \sum_{R,\mathbf{S}} \int P(\mathbf{O}, R, \mathbf{S} \mid \Theta, G) P(\Theta \mid G) \, \mathrm{d}\Theta$$

Learning stage

Given observed data $\mathbf{O}$, find structure $\mathbf{G}$ that maximizes:

$$P(G \mid \mathbf{O}) = \frac{P(\mathbf{O} \mid G)P(G)}{P(\mathbf{O})}$$

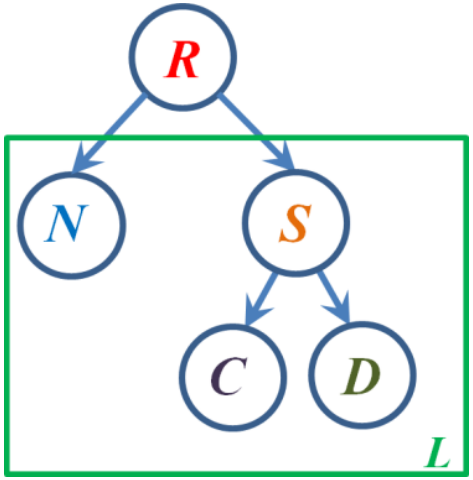
Assuming uniform prior over structures, maximize **marginal likelihood**:

$$P(\mathbf{O} \mid G) \approx P(\mathbf{O}^* \mid G) \cdot \frac{P(\mathbf{O} \mid G, \tilde{\Theta}_G)}{P(\mathbf{O}^* \mid G, \tilde{\Theta}_G)}$$

(Cheeseman-Stutz approximation)

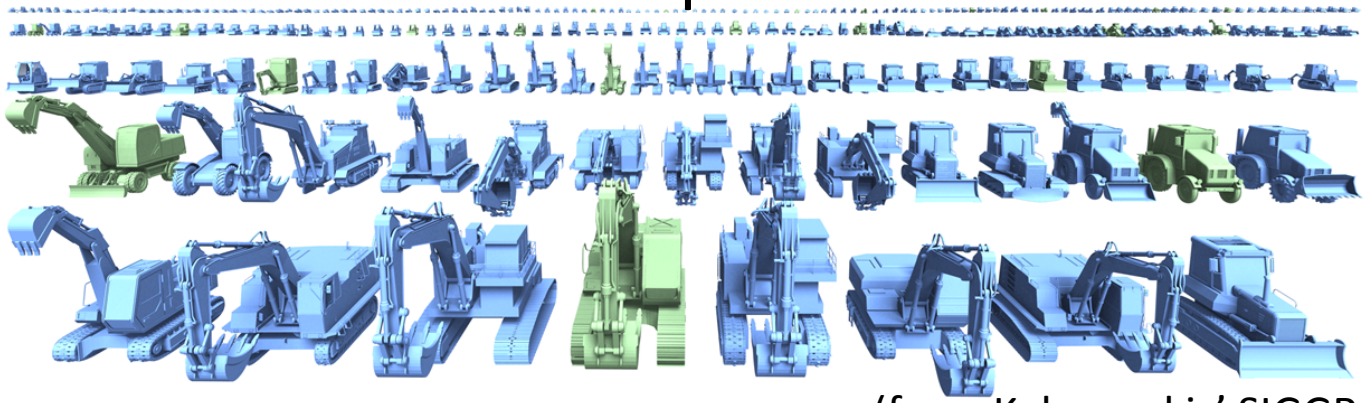
Synthesis Stage

Probabilistic model



Synthesis stage

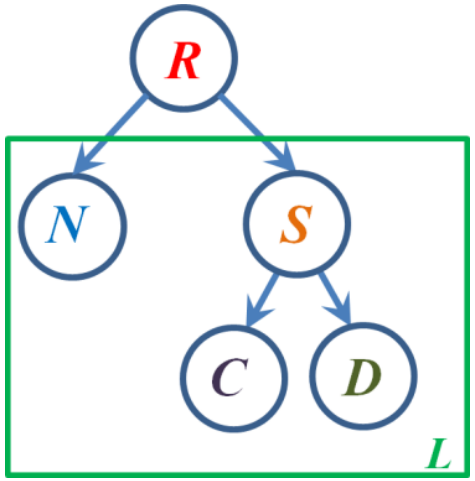
Output

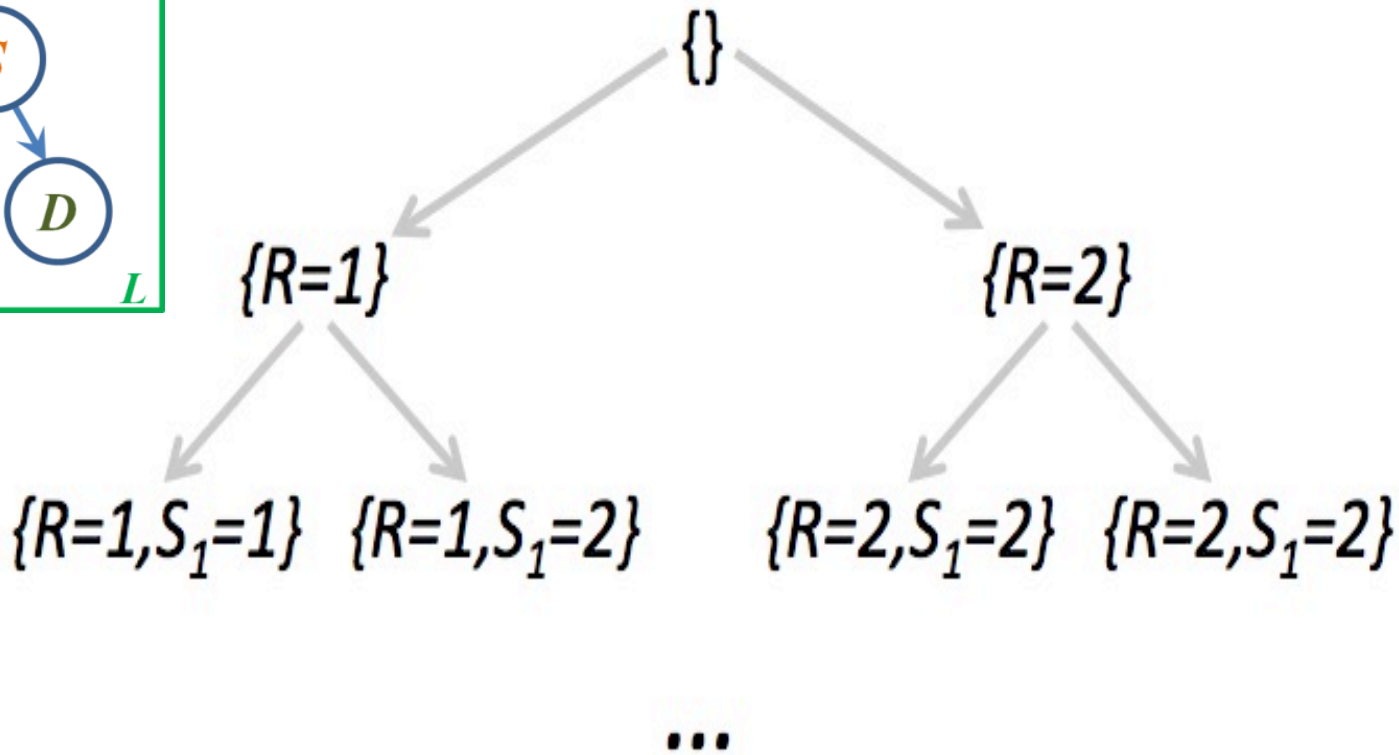


(from Kalogerakis' SIGGRAPH slides)

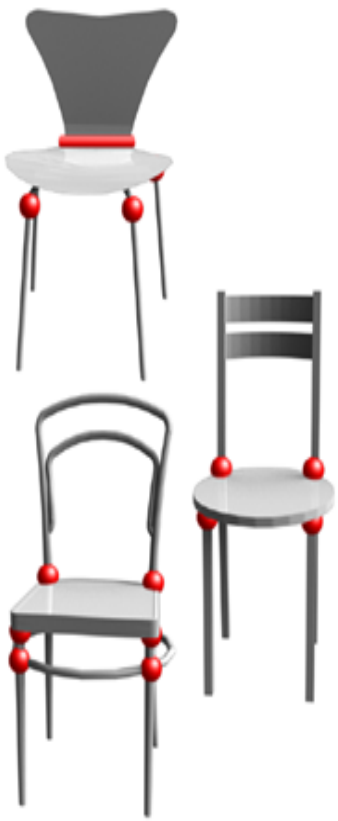
Synthesizing a set of components

Enumerate high-probability instantiations of the model





Optimizing component placement



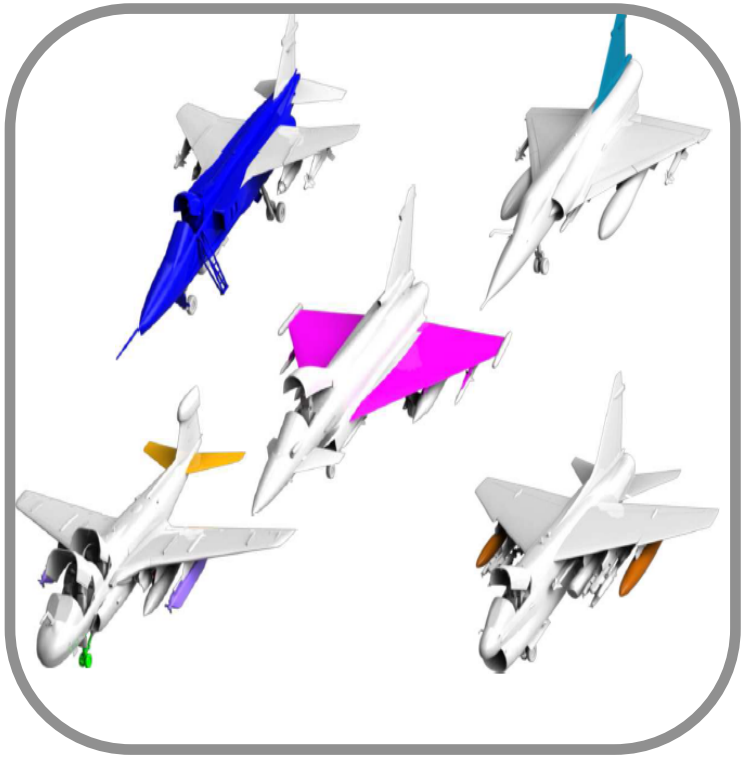
Source shapes

Unoptimized new shape

Optimized new shape

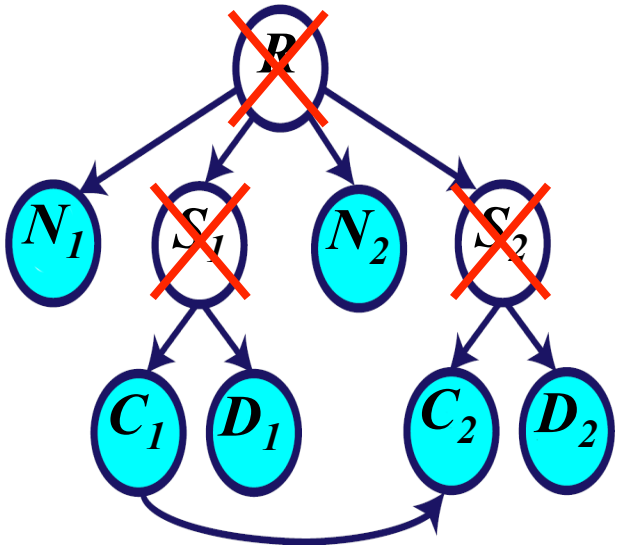
Results

New shape

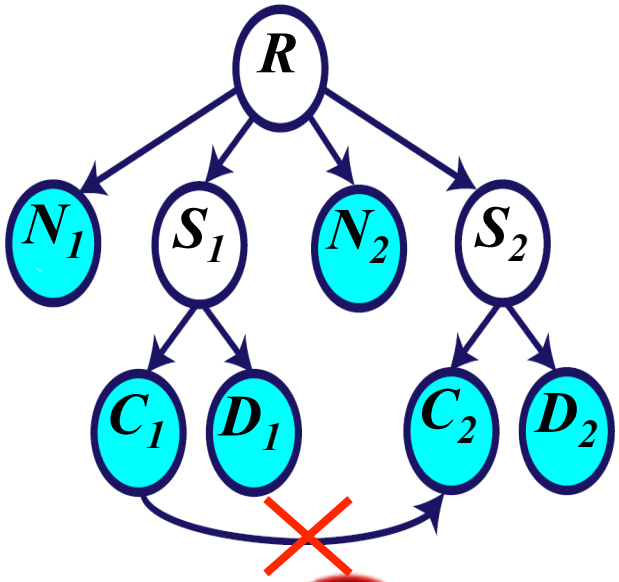


Source shapes
(colored parts are selected for the new shape)

Results of alternative models: no latent variables



Results of alternative models: no part correlations



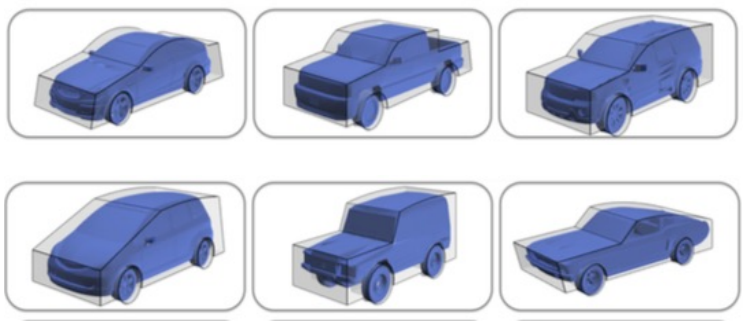
Papers to be covered in this lecture

Shape synthesis

- A Probabilistic Model for Component-based Shape Synthesis [Kalogerakis et al. 12]
- Co-constrained Handles for Deformation in Shape Collections [Yumer et al. 14]

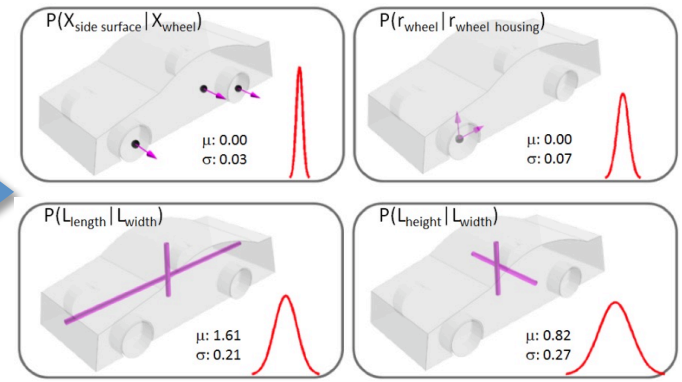
Co-Constrained Handles for Shape Deformation

Input



Learning stage

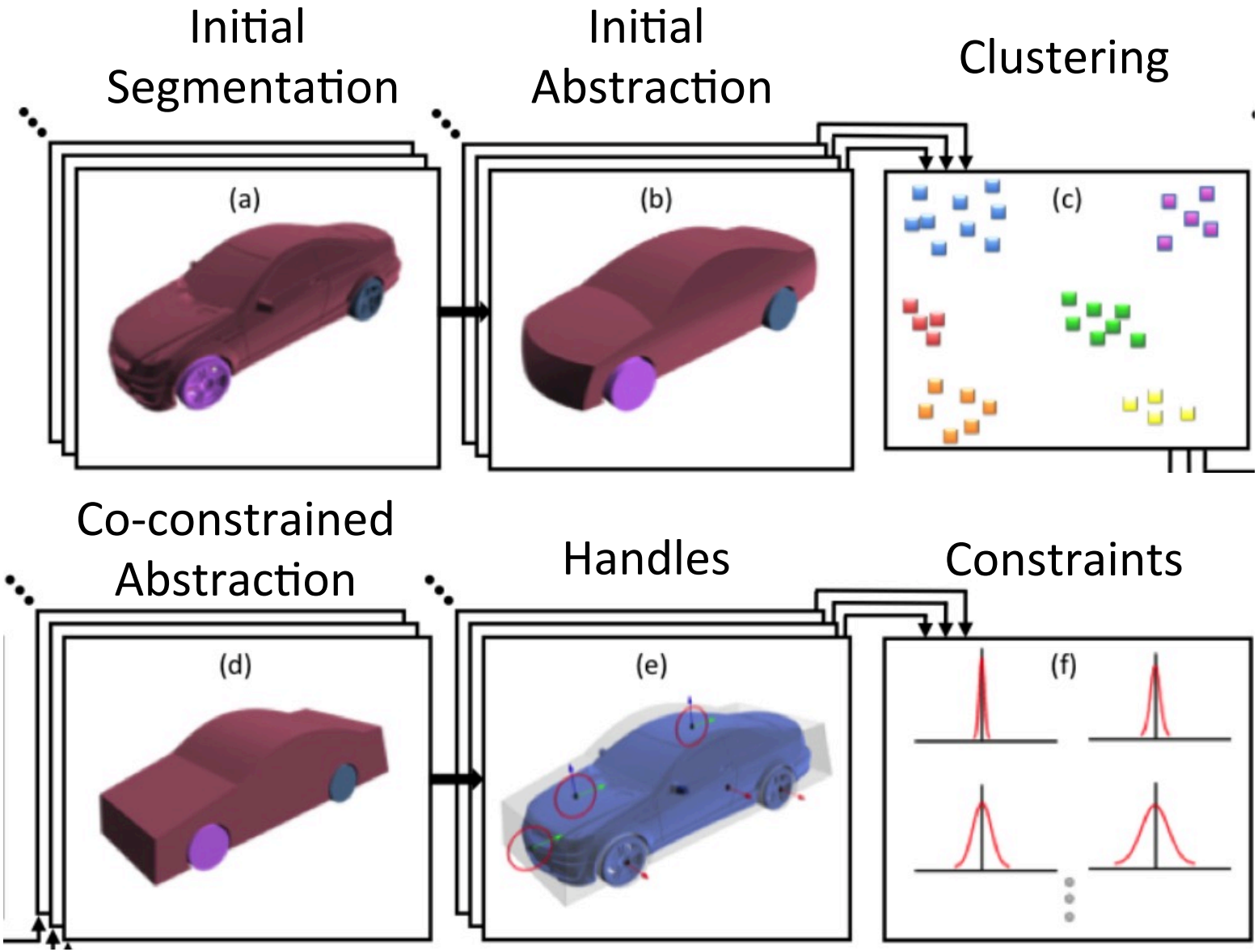
Constraints



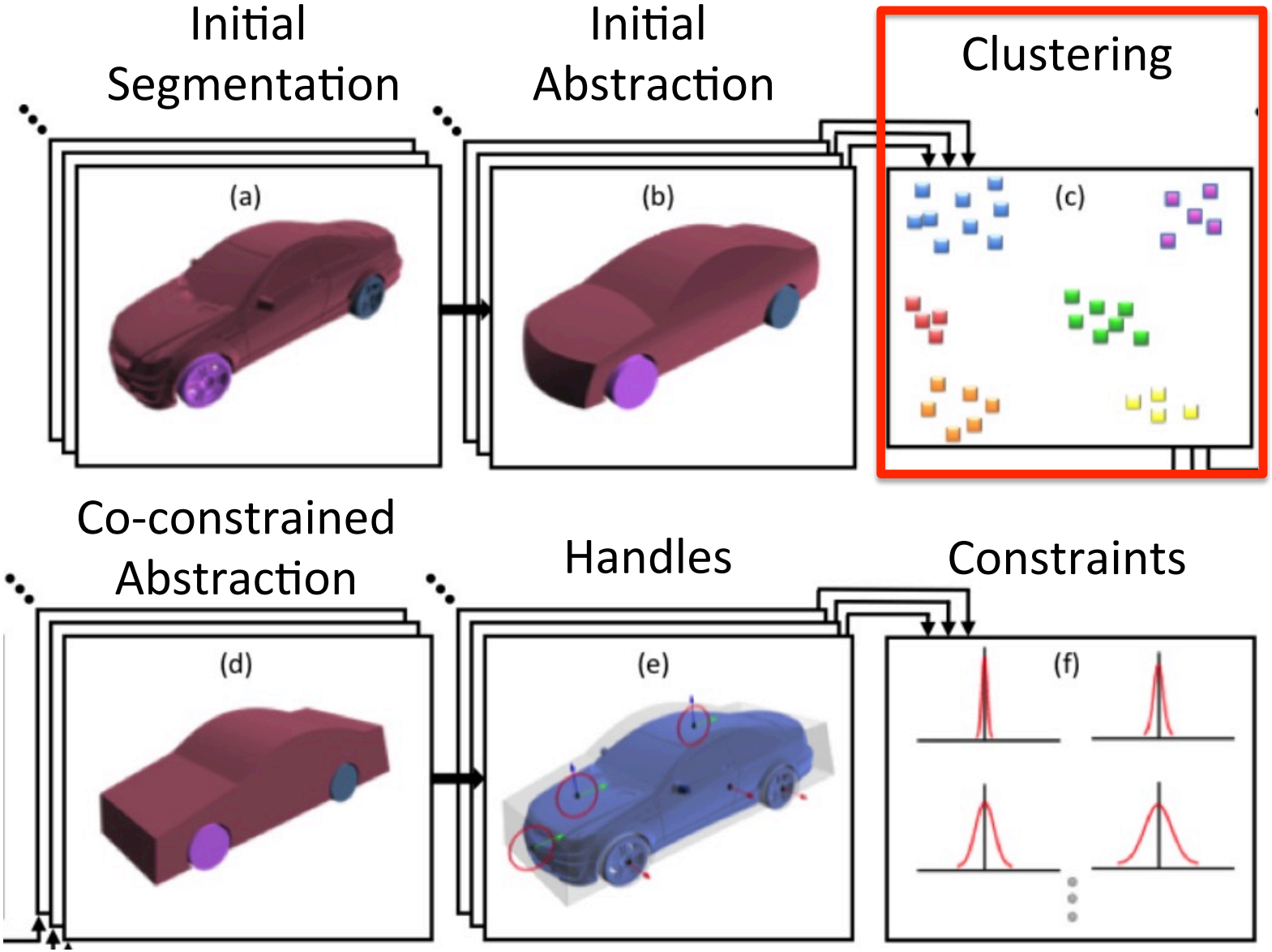
Synthesis stage

Output

Pipeline


Initial Segmentation
Initial Abstraction
Clustering
(a)
(b)
(c)
Co-constrained Abstraction
Handles
Constraints
(d)
(e)
(f)

Pipeline


Initial Segmentation
Initial Abstraction
Clustering
(a)
(b)
(c)
Co-constrained Abstraction
Handles
Constraints
(d)
(e)
(f)

Clustering

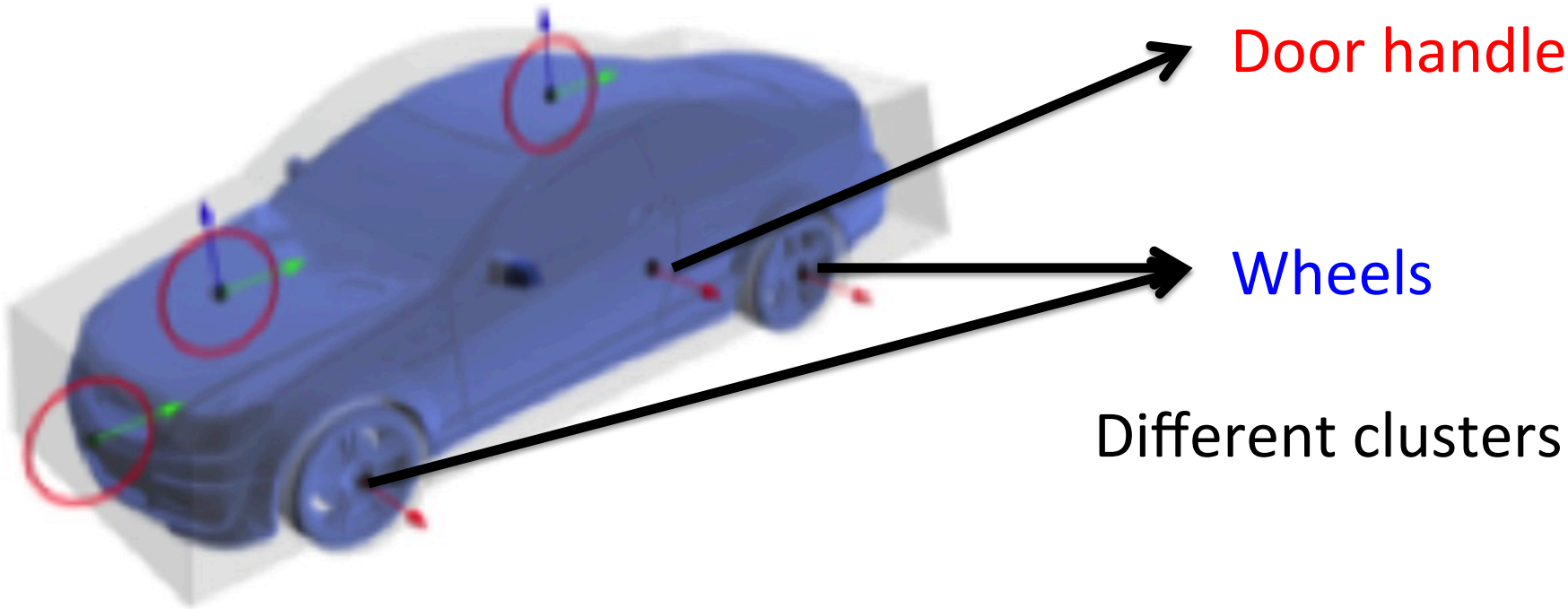
- Seed surface clustering
- Global Clustering

Clustering

- Seed surface clustering
  - Aligning semantically similar/identical segments
- Global Clustering
  - Creating larger clusters that join the surfaces from different segments
  - Based on geometric features of the surfaces, including PCA basis, normal and curvature signature of the surface.

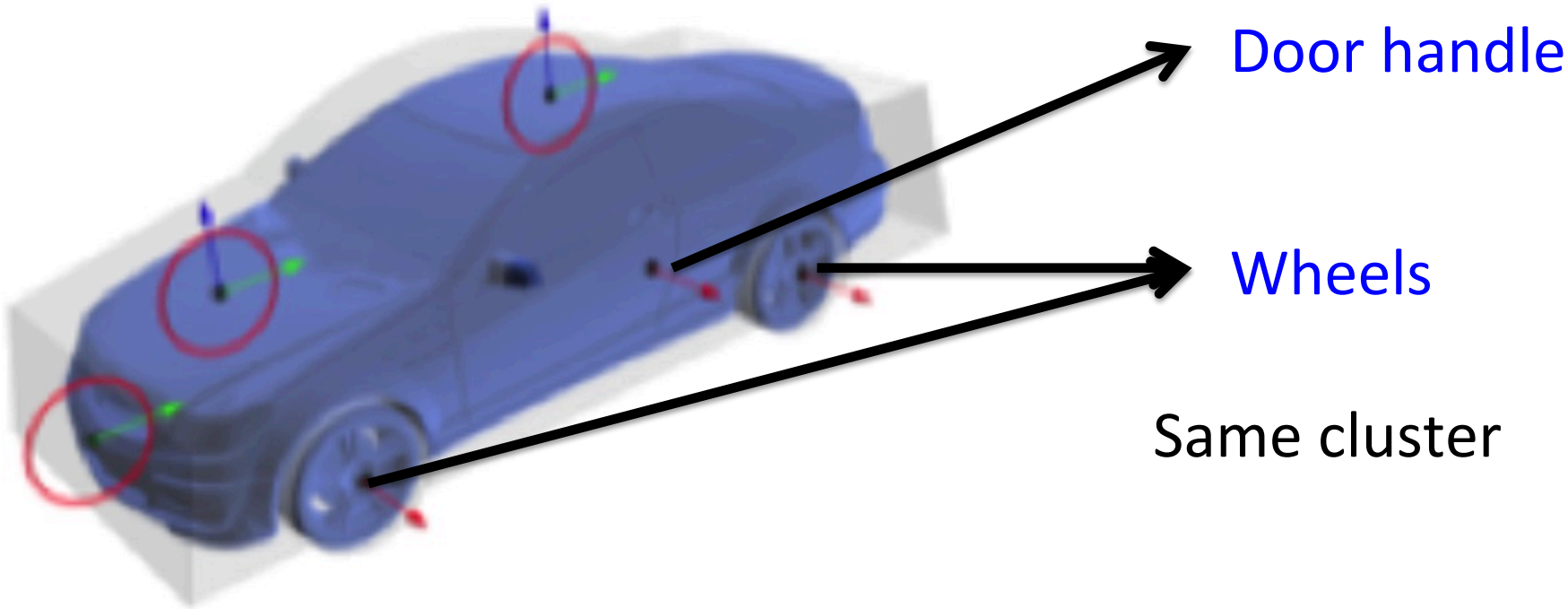
Clustering

Seed clustering

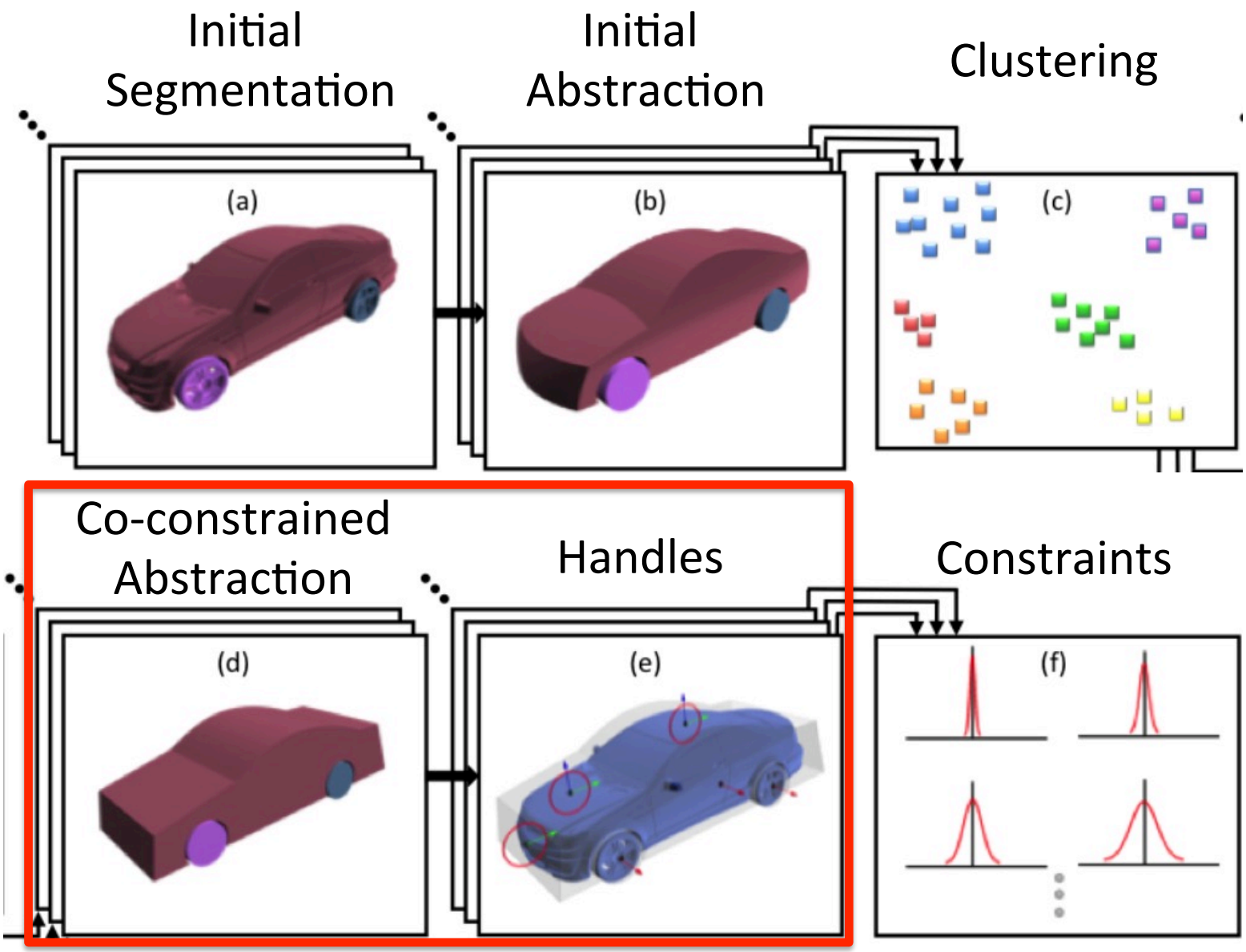


Clustering

Global clustering



Pipeline

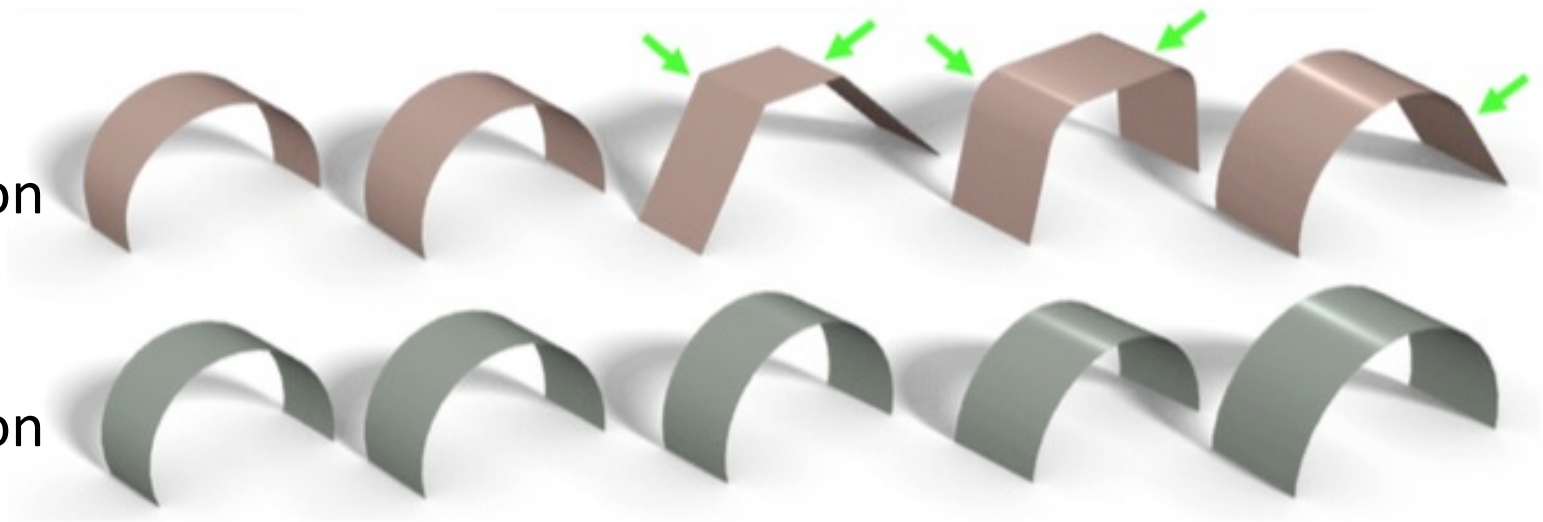

Initial Segmentation
Initial Abstraction
Clustering
(a)
(b)
(c)
Co-constrained Abstraction
Handles
Constraints
(d)
(e)
(f)

Co-constrained abstraction

Planer, spherical, or cylindrical?

Initial abstraction

Co-abstraction

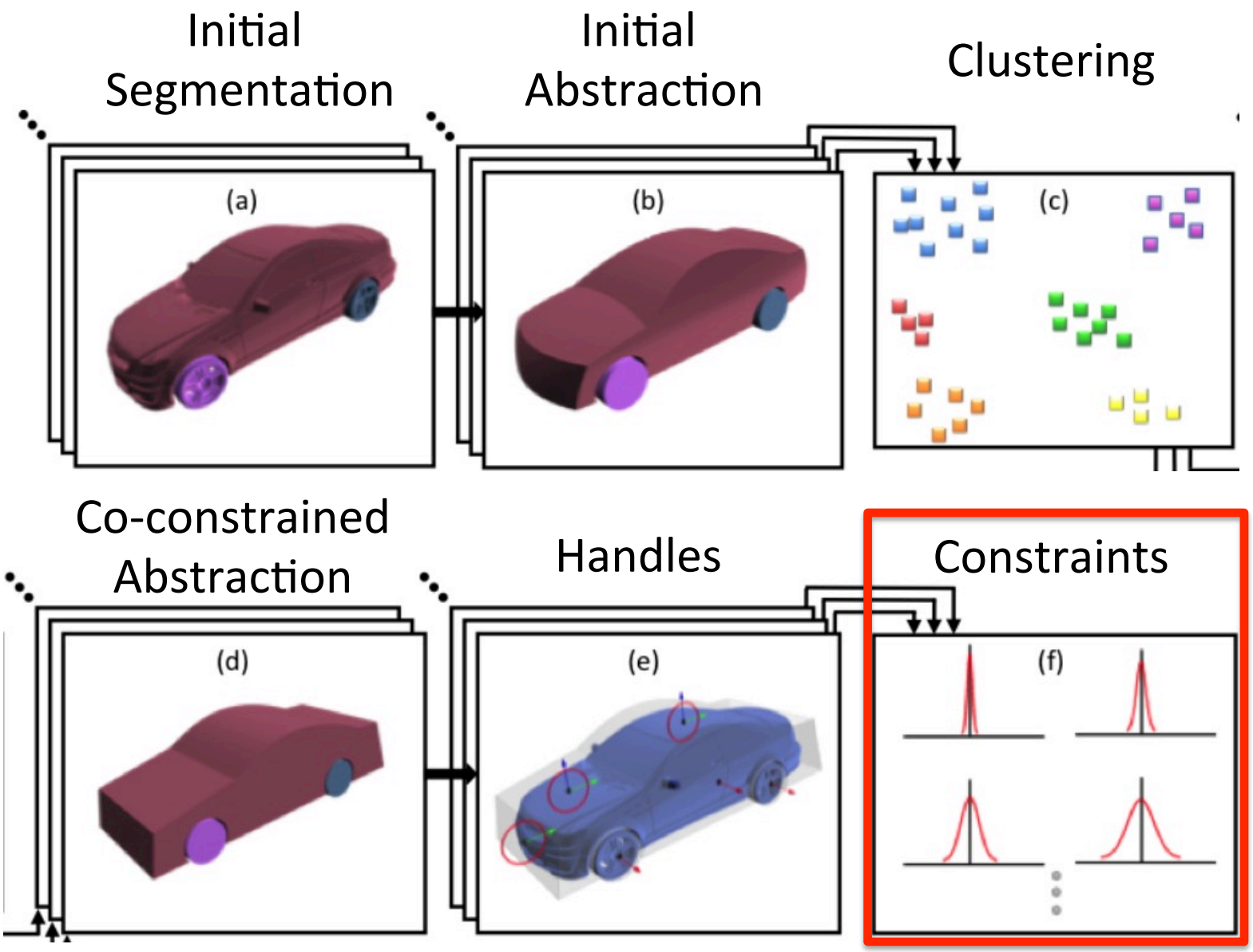


Identify constraints of DOFs

Identify as constraints if variance is below certain threshold

- Translation
- Rotation
- Scaling

Pipeline


Initial Segmentation
Initial Abstraction
Clustering
(a)
(b)
(c)
Co-constrained Abstraction
Handles
Constraints
(d)
(e)
(f)

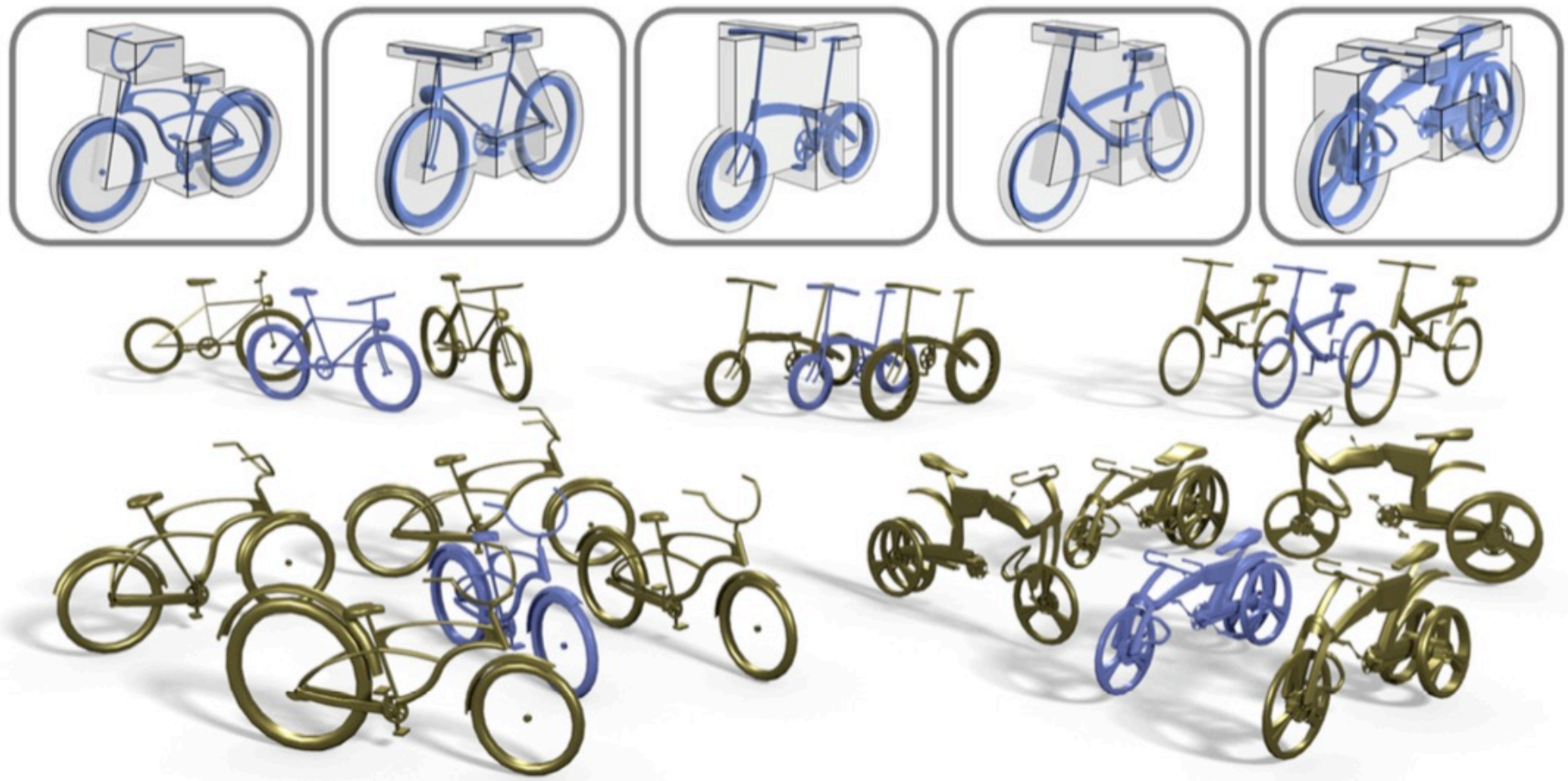
Correlation among deformation handles

Estimate the conditional feature probability

$$\mathbb{P}(k_i|k_j) = \mathbb{P}(k_i, k_j)/\mathbb{P}(k_j)$$

where k takes values from: $t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, s_x, s_y, s_z, r, l.$

Results



Summary

3D shape from image or video

- Using silhouettes, colors or features (e.g. HOG) in image
- Using voxel occupancies, aspect parts, level set, or meshes to represent 3D shapes
- Either generative model, shared shape space, or discriminative model

Summary

Shape synthesis

- Generative model based on semantic parts, which models styles also.
- Modeling correlation between geometric features.