Enforcing View Consistency Using Latent Configurations for 3D Vision Tasks

Arjun Karpur

Supervised by: Dr. Qixing Huang
Why vision?
Success in 2D Vision
The need for 3D Vision

Inferring 3D info from input visuals

Ball

Radius?

Ball

Distance to camera?
3D Vision is not as easy...

**Key assumption:** underlying object or scene is the same across all cameras/views
Methods of gathering 3D ‘images’

Tradeoff between accuracy, efficiency, and cost

Depth scanning hardware
[Levoy et al. 2000]
Learning from synthetic 3D data

[Chang et al. 2015]

Princeton ModelNet

[Wu et al. 2015]
Problems when applying to the real world

vs

Domain gap!
Examples of current domain adaptation techniques in 3D Vision

[Su et al. 2015]
Our proposal

• **Task**: predict 3D geometric structure from a single image

• **Approach**:
  
  • Learn task on synthetic object models (source domain)

  • Leverage the same multi-view assumptions used in gathering 3D images to transfer knowledge to real world images (target domain) in an unsupervised manner

• **Key idea**:

  Geometric predictions from images of the *same object/scene* should be *identical*, up to some rotation and translation
Using multiple views

• Images taken from **varying viewpoints**

• Focused on the **same object or scene**

• Arises **naturally**, found easily online (e.g., YouTube, Flickr, Amazon product pages)

• In our case, no pose information or labels needed

[Xiang et al. 2014]
Latent variables for added benefits

- **Latent variable**: intermediate component, not part of output but assists in optimization

- **Latent configuration**: latent variable representing 3D geometric data in canonical pose for a given image sequence

- Benefits
  - View consistency in **linear time**
  - **Align** latent configurations to source configurations for **extra regularization**
Our view consistency constraint

Source Domain

Target Domain

Output Alignment

Supervised Loss

View Consistency

Green: source ground truths
Blue: predictions
Red: latent configurations
Configuration distance metric

- Given configurations \( X, Y \)
- Need a way to compare similarity between two configurations with respect to task
  - Comparisons between labels, predictions, and latents
- Properties of \( d(X, Y) \)
  - Differentiable
  - Pose-invariant (or camera pose info available)
  - Ideally, efficient to compute (i.e., closed form)
Regularizing Terms

\[
\begin{align*}
\min_{\theta, \{M_i\}} \quad & f_{\text{labeled}} + \lambda f_{\text{view}} + \mu f_{\text{align}}. \\
\end{align*}
\]

Main contributions
Regularizing Terms

\[
\min_{\theta, \{M_i\}} f_{\text{labeled}} + \lambda f_{\text{view}} + \mu f_{\text{align}}.
\]

\[
f_{\text{labeled}} = \frac{1}{|\mathcal{I}|} \sum_{I \in \mathcal{I}} \mathcal{L}(G_{\theta}(I), Y_I)
\]

Compare source predictions to labels
Regularizing Terms

\[
\min_{\theta, \{M_i\}} f_{\text{labeled}} + \lambda f_{\text{view}} + \mu f_{\text{align}}.
\]

\[
f_{\text{labeled}} = \frac{1}{|\mathcal{I}|} \sum_{I \in \mathcal{I}} \mathcal{L}(G_\theta(I), Y_I)
\]

\[
f_{\text{view}} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{I}_i|} \sum_{I_{ij} \in \mathcal{I}_i} d(G_\theta(I_{ij}), M_i)
\]

Compare target predictions to latent configurations
Regularizing Terms

\[
\min_{\theta, \{M_i\}} f_{\text{labeled}} + \lambda f_{\text{view}} + \mu f_{\text{align}}.
\]

\[
f_{\text{labeled}} = \frac{1}{|\mathcal{I}|} \sum_{I \in \mathcal{I}} \mathcal{L}(G_{\theta}(I), Y_I)
\]

\[
f_{\text{view}} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{I}_i|} \sum_{I_{ij} \in \mathcal{I}_i} d(G_{\theta}(I_{ij}), M_i)
\]

\[
f_{\text{align}} = \frac{1}{N} \sum_{i=1}^{N} \min_{I \in \mathcal{I}} d(M_i, Y_I) + \frac{1}{|\mathcal{I}|} \sum_{I \in \mathcal{I}} \min_{1 \leq i \leq N} d(M_i, Y_I)
\]

Chamfer distance to compare distributions of latents and labels
Regularizing Terms

Given loss function (parameterized by network weights and latent configurations), optimize!
Initialization

\[
\min_{\theta,\{M_i\}} f_{\text{labeled}} + \lambda f_{\text{view}} + \mu f_{\text{align}}.
\]

\[
\theta^{(0)} = \min_{\theta} f_{\text{labeled}} = \text{optimal network weights for predicting from source domain}
\]

\[
M_i^{(0)} = \text{best initial prediction, given subset } i
\]

\[\rightarrow \text{Define ‘best’ in terms of distance to source label distribution}\]
Initializing the latent configurations

A visual example...

(3 x d)-space projected to 2D

Initial latent

= target prediction

= source ground truth
Optimization

\[
\text{minimize } f_{\text{labeled}} + \lambda f_{\text{view}} + \mu f_{\text{align}}.
\]

Use alternating minimization!
Alternating minimization - network weights

\[
\theta^{(k+1)} = \underset{\theta}{\arg\min} \frac{1}{|\mathcal{I}|} \sum_{I \in \mathcal{I}} \mathcal{L}(G_{\theta}(I), Y_I) + \frac{\lambda}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{I}_i|} \sum_{I \in \mathcal{I}_i} d(G_{\theta}(I), M_i^{(k)}).
\]

Both $\mathcal{L}$ and $d$ are differentiable, optimize using back propagation
Alternating minimization - latent configurations

\[
\text{minimize}_{\theta, \{M_i\}} f_{\text{labeled}} + \lambda f_{\text{view}} + \mu f_{\text{align}}.
\]

\[
\{M_i^{(k+1)}\} = \text{argmin}_{\{M_i\}} \frac{\mu}{|I|} \sum_{I \in I} \min_{1 \leq i \leq N} d(M_i, Y_I)
\]

\[
+ \frac{1}{N} \sum_{i=1}^N \left( \frac{\lambda}{|I_i|} \sum_{I \in I_i} d(G_{\theta(k)}(I), M_i) + \mu \min_{I \in \overline{I}} d(M_i, Y_I) \right)
\]

\[
\hat{I}(i) = \text{argmin}_{I \in \overline{I}} d(M_i^{(k)}, Y_I), \quad \hat{i}(I) = \text{argmin}_{1 \leq i \leq N} d(M_i^{(k)}, Y_I)
\]
Alternating minimization - latent configurations

\[
\text{minimize } \theta, \{M_i\} \ f_{\text{labeled}} + \lambda f_{\text{view}} + \mu f_{\text{align}}.
\]

\[
M_i^{(k+1)} = \arg\min_{M_i} \frac{\mu}{|\mathcal{I}|} \sum_{I \in \mathcal{I}_i, I^*(I) = i} d(M_i, Y_I) \\
+ \frac{1}{N} \left( \frac{\lambda}{|\mathcal{I}_i|} \sum_{I \in \mathcal{I}_i} d(G_{\theta(k)}(I), M_i) + \mu d(M_i, Y_{\hat{i}(i)}) \right)
\]

Given fixed pairs, can optimize each latent independently (weighted average)
Optimizing network weights and latents iteratively

Revisiting our toy example...

Step 1: Optimize network weights
Step 2: Optimize latent configurations
Step 3: Optimize network weights again!

...repeat until converges

(3 x d)-space projected to 2D
View Consistency for 3D Keypoint Estimation
Experimental setup

- ModelNet (source domain) to...
Real image domain datasets

[A Large Dataset of Object Scans]

[Choi et al. 2016]
Keypoint representation

- Keypoint configuration as ordered matrix $X \in \mathbb{R}^{3 \times d}$
  - Fixed number of keypoints
  - Centered around origin

- Pose-invariant distance function:
  $$d(X, Y) = \min_{R \in SO(3)} \left\| RX - Y \right\|_{F}^{2}$$
  - Admits closed form method of evaluation and differentiation [BK Horn 1987]
  - Automatically solves for rotation $R$
Quantitative results

![Graph showing quantitative results with different lines for Default, Ours, DropAlign, DropView, Re-initialization, ADDA, and Supervised. The x-axis represents percentage of keypoints, and the y-axis represents average distance error percentage of 3D bounding box diagonal.](image-url)
Qualitative results - chairs
Qualitative results - motorcycles
Qualitative results - extensions
View Consistency for Shape Reconstruction
Shape representation

- Represent shape as voxels $X \in \mathbb{R}^{d \times d \times d}$
  - Voxelize ground truth models and center around origin
  - Crop Redwood real meshes around object
  - Predict probabilities and bin over/under threshold to [0,1]

- Use Binary Cross Entropy loss function as our task-specific loss function $\mathcal{L}$

- Use L2-Loss as our distance metric $d(X, Y)$
  - To address pose-invariance, always predict in canonical pose
Datasets

Source domain: ShapeNet
Target domain: Redwood
TL Embedding for Shape Reconstruction

[Image: Diagram showing the process of TL Embedding for Shape Reconstruction.]

[Text: TL Embedding for Shape Reconstruction]

[Reference: Girdhar et al. 2016]
Qualitative results - example

Input Image  Baseline pred  Our pred  Supervised pred  Ground truth
Qualitative results
Consistent predictions across views
Quantitative results

Intersection over Union accuracy (in percentage)

<table>
<thead>
<tr>
<th>Target-Metric</th>
<th>TL-Default</th>
<th>TL-View</th>
<th>TL-Supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShapeNet</td>
<td>41.1</td>
<td>40.7</td>
<td>28.6</td>
</tr>
<tr>
<td>RedwoodRGB</td>
<td>25.7</td>
<td>27.7</td>
<td>28.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target-Metric</th>
<th>TL-Default-Train</th>
<th>TL-View-Train</th>
<th>TL-Supervised-Train</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShapeNet</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RedwoodRGB</td>
<td>27.5</td>
<td>28.7</td>
<td>-</td>
</tr>
</tbody>
</table>
Domain adaptation via unsupervised view consistency for

3D Keypoint Estimation

Shape Reconstruction

... Arbitrary geometric information
Future work

• The future of massive image data: depth scans
  • Jumpstart multi-view reconstruction with a good initial guess

• Conversely, remove need for specialized hardware. Slim down 3D systems
  • Using prior knowledge in single-view 3D vision inference
Acknowledgements

Thank you to all my family, friends, and peers for the continuous support!

- Dr. Qixing Huang
- Xingyi Zhou
- Graphics & AI Lab
- Dr. Alex Huth, Dr. Emmett Witchel (committee)