All Graphs Lead to Rome: Learning Geometric and Cycle-Consistent Representations with Graph Convolutional Networks

Credit: Stephen Phillips, Kostas Daniilidis

GRASP Laboratory, University of Pennsylvania

Presenter: Fuwen Tan

https://qdata.github.io/deep2Read
A simplified image (feature) matching problem

- Match the feature points across multiple images.
  - N images
  - M repeatable feature points, with known coordinates, and initial descriptors (e.g. SIFT)
    - A simplified version of the general image matching problem
  - Goal → assignment problem: find a one-to-one mapping between the points in each image pair, and these mappings are globally (cycle) consistent.

Figure: Multiple image matching.
Soft assignment by learning an embedding vector for each point.

\[ \mathcal{U} = \{\star, \bigcirc, \blacksquare\} \quad \mathcal{E} = \{I_1, I_2, I_3\} \]

**Figure:** Illustration of the embedding the model is training to learn. The colors used here are inconsistent.
Adjacent Matrix

- $E = \{E_i\}_{i=1}^{N \cdot M}$
- Ideal adjacent matrix of all features
  - $\hat{A} \in \mathbb{R}^{(N \cdot M) \times (N \cdot M)}$
  - $\hat{A}_{ij} = E_i E_j^T \in \{0, 1\}$
  - $i$ and $j$ are connected if they represent the same 3D point.
Graph Convolution

\[ E^{(l+1)} = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} E^{(l)} W^{(l)} \right) \]  

- \( E^0 \in \mathbb{R}^{(N \cdot M) \times 128} \): hand-crafted features
  - L2-normalized SIFT + log scale SIFT + \textit{calibrated x-y position} + orientation.
- \( \tilde{A} = A + I \in \mathbb{R}^{(N \cdot M) \times (N \cdot M)} \)
- \( \tilde{D}_{ii} = \sum_j \tilde{A}_{ij} \)
- \( A \): a soft adjacent matrix computed from \( E^0 \). And I guess they keep it fixed for each training sample.
Cycle consistency loss [4]

\[ L(\tilde{A}, EE^T) = | \tilde{A} - EE^T | \] (2)

- E: final embedding
- The paper uses A in (2), but I guess it should be \( \tilde{A} \)
- The name (cycle consistency loss) is from another optimization-based paper [4]
- Why not use the ground-truth adjacent matrix?
  - The paper claims the method is unsupervised.
Geometric consistency loss

\[ L(E) = \sum_{i,j} <E_i, E_j> G_{ij} \quad (3) \]

\[ M = R_{c(i)}^T [T_{c(j)} - T_{c(i)}] \times R_{c(j)} \quad (4) \]

\[ G_{ij} = |X_i^T M X_j| \quad (5) \]

- \( X_i \in \mathbb{R}^3 \): homogeneous normalized image coordinates for \( E_i \)
- \( c(i) \): the camera for \( E_i \)
- \( R_{c(i)}, T_{c(i)} \): the camera pose for \( c(i) \)
- \( M \in \mathbb{R}^{3\times3} \): essential matrix [3], wiki
Unsupervised?

- $G = \{G_{ij}\}$ is very similar with $1 - \hat{A}$;
- $\hat{A}$: ground-truth adjacent matrix
Figure: Illustration of the approach of this work.
Experiment 1: synthetic graph dataset

1. $M$ points (unknown $M$) in 3D, with a ground-truth embedding vector for each point;
2. $\tilde{A}$: random permutation matrix (ground-truth adjacent matrix) + gaussian noise
3. From (1) (2), we can have the ground-truth $E$
4. $E^0$: $E$ + gaussian noise, which is used as baseline
5. 12 layers GCN with ReLU activation and skip connection (unknown architecture);
6. Just the first (cycle consistency) loss
Experiment 1: observations

1. $\tilde{A}$ is from ground-truth instead of $E^0$

2. The final embedding $E$ is trained using a noisy ground-truth adjacent matrix. Intuitively, $E$ should be better than $E^0$
## Experiment 1: results

<table>
<thead>
<tr>
<th>Method</th>
<th>Same Point Similarities</th>
<th>Different Point Similarities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>$1.00e+0 \pm 0.00e+0$</td>
<td>$0.00e+0 \pm 0.00e+0$</td>
</tr>
<tr>
<td>Initialization Baseline</td>
<td>$5.11e-1 \pm 1.68e-2$</td>
<td>$2.56e-1 \pm 2.06e-1$</td>
</tr>
<tr>
<td>3 Views, Noiseless</td>
<td>$9.96e-1 \pm 7.70e-3$</td>
<td>$1.16e-1 \pm 1.32e-1$</td>
</tr>
<tr>
<td>5 Views, Noiseless</td>
<td>$1.00e+0 \pm 4.15e-4$</td>
<td>$1.22e-1 \pm 1.67e-1$</td>
</tr>
<tr>
<td>3 Views, Added Noise</td>
<td>$9.96e-1 \pm 7.70e-3$</td>
<td>$1.16e-1 \pm 1.32e-1$</td>
</tr>
<tr>
<td>5 Views, Added Noise</td>
<td>$9.89e-1 \pm 2.47e-2$</td>
<td>$7.67e-2 \pm 1.56e-1$</td>
</tr>
<tr>
<td>6 Views, Added Noise</td>
<td>$9.84e-1 \pm 3.16e-2$</td>
<td>$7.46e-2 \pm 1.57e-1$</td>
</tr>
<tr>
<td>3 Views, 5% Outliers</td>
<td>$9.29e-1 \pm 1.79e-1$</td>
<td>$1.41e-1 \pm 1.48e-1$</td>
</tr>
<tr>
<td>3 Views, 10% Outliers</td>
<td>$9.27e-1 \pm 1.79e-1$</td>
<td>$1.40e-1 \pm 1.51e-1$</td>
</tr>
</tbody>
</table>

### Table: Results on Synthetic correspondence graphs. The ‘Same Point Similarities’ column is the similarities for true corresponding points, while the ‘Different Point Similarities’ is for points that do not correspond. Losses tested against ground truth correspondence graph adjacency matrices. Our method was not trained on ground truth correspondences but using unsupervised methods.

Experiment 2: Rome 16K Graph Dataset [2]

1. A dataset consists of 16000 images of historical sites in Rome, with corresponding annotations;
2. 80 points (M = 80) in 3D;
3. Image triplets and quadruplets (N = 3 or 4)
4. 12-layers GCN, skip connection between the 6th and 12th layers, still unknown architecture
5. Only evaluate the L1 and L2 losses, no evaluation on the accuracy.
Table: Results on Rome16K Correspondence graphs. Our method was not trained on ground truth correspondences but using unsupervised methods and geometric side losses. As our method gives soft labels, we use cannot use precision or recall as is standard in testing cycle consistency [5]. Thus we test against ground truth correspondence graph adjacency matrices computed from the bundle adjustment output.
Conclusion

- Over-claiming;
- The formulation seems odd;
- Experiments cannot demonstrate the effectiveness of the method;
- The figures are not well explained in the paper.

