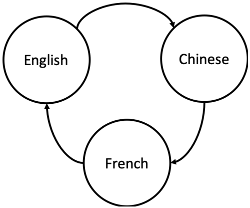
Condition Number for Joint Optimization of Cycle-Consistent Networks

Leonidas Guibas[1] , Qixing Huang[2] and Zhenxiao Liang[2]

[1]Stanford University, [2]University of Texas at Austin

Basic Idea and Cycle Consistency

- We can employ cycle consistency to improve the performance of multiple neural networks among several domains when the transformations form some cycles.
- Applications: translation, shape matching, CycleGAN, 3D model representations etc.



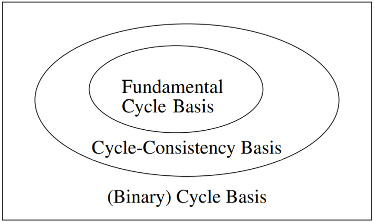
The choice of cycles used to enforce cycle consistency is important when there are many domains.

Mapping Graph and Cycle Bases

- A mapping graph is a directed graph $\mathcal{G}_f = (\mathcal{V}, \mathcal{E})$ such that each node $u \in \mathcal{V}$ is associated with a domain $\mathcal{D}_u$ and each edge $(u, v) \in \mathcal{E}$ with a function $f_{uv} : \mathcal{D}_u \to \mathcal{D}_v$.
- The cycle bases can be defined in several manners depending on what binary operator of cycles is used to compose new cycles.
- The most common bases are binary cycle bases and fundamental cycle bases.

Cycle Bases

- It has been known that there always exist binary cycle bases of size $|\mathcal{E}| - |\mathcal{V}| + 1$.
- In particular, a fundamental cycle bases can be easily constructed from a spanning tree on $\mathcal{G}$.
- Not all binary bases are cycle-consistent.



Cycle-Consistency Bases

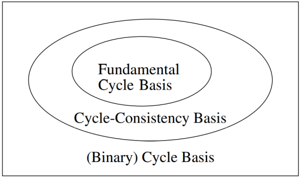
- A mapping graph $\mathcal{G}_f$ is called cycle consistent if the composition of f along each cycle in $\mathcal{G}_f$ is identity, i.e.,

$$f_{u_k u_1} \circ f_{u_{k-1} u_k} \circ \cdots \circ f_{u_2 u_3} \circ f_{u_1 u_2} = I.$$

- The number of cycles in a graph can be exponentially large. It is impossible to enforce consistency on all cycles directly in large graphs.
- A cycle bases $\mathcal{B} = \{C_1, \ldots, C_{|\mathcal{B}|}\}$ is cycle-consistent if cycle-consistency is guaranteed over all cycles in $\mathcal{G}$ for *any* function family f whenever f is cycle-consistent along cylces in $\mathcal{B}$.

Cycle-Consistency Bases

- Fundamental bases always work but not perfect.
- Intuitively it will be harder to optimize f along a longer cycle. Fundamental bases come from the spanning trees of graphs so that can contain many long cycles.



Simple Case of Translation Synchronization

Specifically we consider the translation functions $f_{ij}(x) := x + t_{ij}$ where t_{ij} is parameters to be optimized. Suppose t_{ij}^0 is the initial parameter.

Loss Function:

$$\min_{\{t_{ij},(i,j)\in\mathcal{E}\}} \sum_{(i,j)\in\mathcal{E}_0} (t_{ij} - t_{ij}^0)^2 + \sum_{c=(i_1\cdots i_k i_1)\in\mathcal{C}} w_c \Big(\sum_l t_{i_l i_{l+1}}\Big)^2. \tag{1}$$

We hope the final t_{ij} are close to $t_{ij}^{(0)}$ and keep the cycle consistency.

Condition Number for Translation Case

(1) can be rewritten in matrix form:

$$\min_{\boldsymbol{t}} \ \boldsymbol{t}^T H \boldsymbol{t} - 2\boldsymbol{t}^T \boldsymbol{t}^0 + \|\boldsymbol{t}^0\|^2, \quad (2)$$

$$H := \sum_{e \in \mathcal{E}^0} \boldsymbol{v}_e \boldsymbol{v}_e^T + \sum_{c \in \mathcal{C}} w_c \boldsymbol{v}_c \boldsymbol{v}_c^T.$$

- This quadratic optimization problem is generally relevant to condition number $\kappa(H) = \lambda_{\max}(H)/\lambda_{\min}(H)$.
- The deviation between the optimal solution $\boldsymbol{t}$and the ground truth $\boldsymbol{t}^{gt}$ ground truth solution is

$$\|\boldsymbol{t}^* - \boldsymbol{t}^{gt}\| \leq \frac{1}{\lambda_{\min}(H)} \|\boldsymbol{t}^0 - \boldsymbol{t}^{gt}\|$$

where $\boldsymbol{t}^0$ is the initialization translation vector.

Sampling Process (Step I - $\mathcal{C}_{sup}$ generation)

We construct $\mathcal{C}_{sup}$ by computing the breadth-first spanning tree $\mathcal{T}(v_i)$ rooted at each vertex $v_i \in V$. The resulting $\mathcal{C}_{sup}$ has two desired properties:

- The cycles in $\mathcal{C}_{sup}$ are kept as short as possible.
- If $\mathcal{G}$ is sparse, then $\mathcal{C}_{sup}$ contains a mixture of short and long cycles. These long cycles can address the issue of accumulated errors if we only enforce the cycle-consistency constraint along short cycles.

Sampling Process (Step II - Weight Optimization)

We formulate the following semidefinite program for optimizing cycle weights:

$$\min_{w_c \geq 0, s_1, s_2} \quad s_2 - s_1 \tag{3}$$

$$\text{subject to} \quad s_1 I \preceq \sum_{e \in \mathcal{E}^0} \boldsymbol{v}_e \boldsymbol{v}_e^T + \sum_{c \in \mathcal{C}_{sup}} w_c \boldsymbol{v}_c \boldsymbol{v}_c^T \preceq s_2 I$$

$$\sum_{c \in \mathcal{C}_{sup}} |\boldsymbol{v}_c|^2 w_c = \lambda, \qquad w_c \geq \delta, \forall c \in \mathcal{C}_{\min} \tag{4}$$

- (3) enforces H close to an identity matrix.
- $w_c \geq \delta$ for $c \in \mathcal{C}_{\min}$ guaranteed cycles in $\mathcal{C}_{\min}$ taken into account.

Importance Sampling

The semidefinite program described above controls the condition number of H, but it does not control the size of the cycle sets with positive weights.

We seek to select a subset of cycles $\mathcal{C}_{sample} \subset \mathcal{C}_{sup}$ and compute new weights $\overline{w}_c, c \in \mathcal{C}_{sample}$, so that

$$\sum_{c \in \mathcal{C}_{sample}} \overline{w}_c \boldsymbol{v}_c \boldsymbol{v}_c^T \approx \sum_{c \in \mathcal{C}_{sup}} \overline{w}_c \boldsymbol{v}_c \boldsymbol{v}_c^T . \tag{5}$$

Main Results of Sampling

Under mild assumptions, w.h.p we have

$$E[|\mathcal{C}_{sample}|] = L, \quad (6)$$

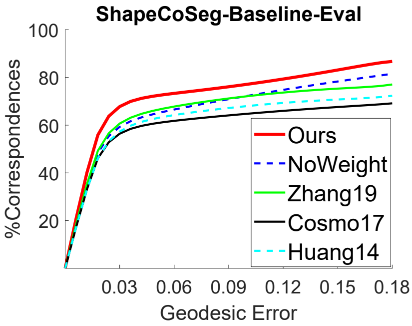
$$E[\sum_{c\in\mathcal{C}_{sample}} \overline{w}_c \mathbf{v}_c \mathbf{v}_c^T] = \sum_{c\in\mathcal{C}_{sup}} w_c \mathbf{v}_c \mathbf{v}_c^T \quad (7)$$

$$||\mathcal{C}_{sample}| - L| \leq O(\log n)\sigma_1 \quad (8)$$

$$\|\sum_{c\in\mathcal{C}_{sample}} \overline{w}_c \mathbf{v}_c \mathbf{v}_c^T - \sum_{c\in\mathcal{C}_{sup}} w_c \mathbf{v}_c \mathbf{v}_c^T\| \leq O(\log n)\sigma_2 \quad (9)$$

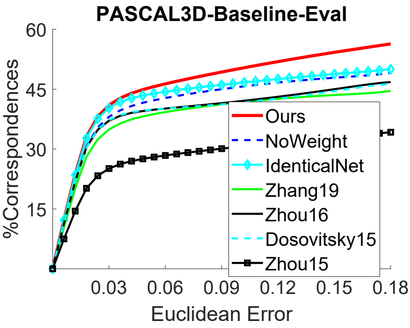
where n is the number of domains and σ_1^2 and σ_2^2 are the unweighted and weighted variances of $|\mathcal{C}_{sample}|$ respectively.

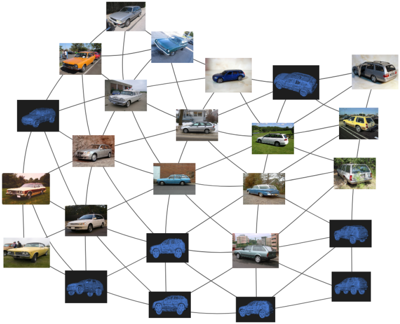
Experimental Results - Consistent Shape Correspondence



- Encode the map from one shape S_i and another shape S_j as a functional map $X_{ij} : \mathcal{F}(S_i) \to \mathcal{F}(S_j)$.
- Considered two shape collections from ShapeCoSeg: Alien (200 shapes) and Vase (300 shapes).
- Construct $\mathcal{G}$ by connecting every shape with $k = 25$ randomly chosen shapes.

Experimental Results - Consistent Neural Networks





- $\mathcal{V}$ represents image objects viewed from similar camera poses.
- Jointly learn the neural networks associated with each edge.