Efficient, General Point Cloud Registration with Kernel Feature Maps

Hanchen Xiong, Sandor Szedmak, Justus Piater

Institute of Computer Science
University of Innsbruck

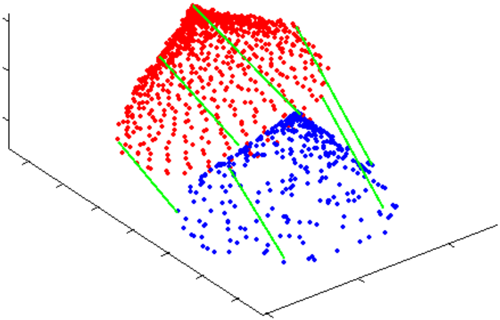
30 May 2013

Outline

1. Background

Problem statement

- 3D point cloud registration
  Given two point clouds $\mathbf{X}_1 = \{x_i^{(1)}\}_{i=1}^{l_1}$, $\mathbf{X}_2 = \{x_j^{(2)}\}_{j=1}^{l_2}$, find the correct correspondences between $x_i^{(1)}$ and $x_j^{(2)}$, based on which two point clouds can be aligned.



Related Work

Registration

- Iteration Closest Point (ICP);
  - match nearest neighbours as correspondences $\rightleftharpoons$ minimize the distances between correspondences
- Gaussian Mixture;
  - fit point clouds to distributions + correlation, L2 distance or kernel methods
- SoftAssign / EM-ICP
  - one-to-many correspondences
  - optimize w.r.t. correspondence matrix $\rightleftharpoons$ optimize w.r.t. transformation.

Related Work, cont.

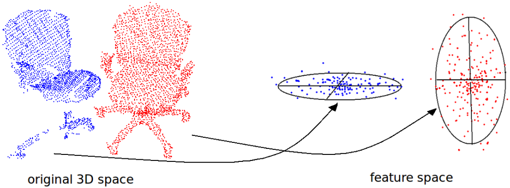
$$\{\mathbf{R}^*, \mathbf{b}^*\} = \arg\min_{\mathbf{R},\mathbf{b}} \sum_{i=1}^{l_1} \sum_{j=1}^{l_2} \left(\mathbf{R}\mathbf{x}_i^{(1)} + \mathbf{b} - \mathbf{x}_j^{(2)} \right)^2 w_{i,j} \tag{1}$$

where $\mathbf{R}, \mathbf{b}$ denote rotation and translation in $\mathbb{R}^3$.

- - ICP: $w_{i,j} \in \{0, 1\}$, determined by shortest-distance criterion;
- - Guassian Mixtures: and $w_{i,j} = \frac{1}{l_1 l_2}$ for all i, j (uniformly);
- - SoftAssign/EM-ICP: $w_{i,j}$ is interpreted as the probability of the correspondence;

2. Transformation in Hilbert Space

Kernel Method & Feature Map



By applying a kernel function on 3D points $K(x_i, x_j)$, a $\mathbb{R}^3 \to \mathcal{H}$ feature map ϕ is implicitly induced:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{2}$$

and $\mathcal{H}$ is called Hilbert space, which is usually much higher or possibly infinite dimensional:

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2/2\sigma^2} \quad \to \quad \phi(x_i) \propto f(\xi) = e^{-\|\xi - x_i\|^2/2\sigma^2} \tag{3}$$

Gaussian in Hilbert Space

mean:

$$\boldsymbol{\mu}_{\mathcal{H}}^{(1)} = \frac{1}{l_1} \sum_{i=1}^{l_1} \phi(x_i^{(1)}) \tag{4}$$

$$\boldsymbol{\mu}_{\mathcal{H}}^{(2)} = \frac{1}{l_2} \sum_{i=1}^{l_2} \phi(x_i^{(2)}) \tag{5}$$

covariance :

$$\mathbf{C}_{\mathcal{H}}^{(1)} = \frac{1}{l_1} \sum_{i=1}^{l_1} \left(\phi(x_i^{(1)}) - \boldsymbol{\mu}_{\mathcal{H}}^{(1)}\right) \left(\phi(x_i^{(1)}) - \boldsymbol{\mu}_{\mathcal{H}}^{(1)}\right)^{\top} \tag{6}$$

$$\mathbf{C}_{\mathcal{H}}^{(2)} = \frac{1}{l_2} \sum_{i=1}^{l_2} \left(\phi(x_i^{(2)}) - \boldsymbol{\mu}_{\mathcal{H}}^{(2)}\right) \left(\phi(x_i^{(2)}) - \boldsymbol{\mu}_{\mathcal{H}}^{(2)}\right)^{\top} \tag{7}$$

Kernel PCA

Assume all points are already centralized:

$$C = \frac{1}{l}\sum_{i=1}^{l}\phi(x_i)\phi(x_i)^\top \tag{8}$$

the none-zero eigenvalue λ_k and corresponding eigenvector $\mathbf{u}_k$ of C should satisfy:

$$\lambda_k \mathbf{u}_k = C\mathbf{u}_k \tag{9}$$

by substituting (8) into (9), we can have:

$$\mathbf{u}_k = \frac{1}{\lambda_k} C\mathbf{u}_k = \sum_{i=1}^{l}\alpha_i^k \phi(x_i) \tag{10}$$

where $\alpha_i^k = \frac{\phi(x_i)^\top \mathbf{u}_k}{\lambda_k l}$. Therefore, all eigenvectors $\mathbf{u}_k$ with $\lambda_k \neq 0$ must lie in the span of $\phi(x_1), \phi(x_2), \ldots.\phi(x_l)$, and (10) is referred to as the dual form of $\mathbf{u}_k$.

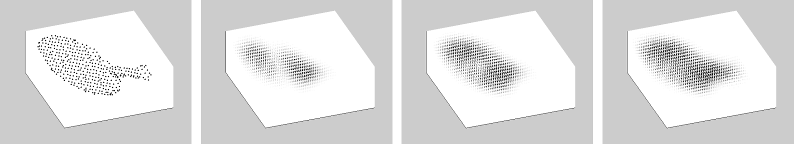
Kernel PCA, cont.

By left multiplying $\sum_{j=1}^{l} \phi(x_j)^\top$ on both sides of equation (9) :

$$
\begin{aligned}
& \sum_{j=1}^{l} \phi(x_j)^\top \lambda_k \mathbf{u}_k = \sum_{j=1}^{l} \phi(x_j)^\top C \mathbf{u}_k \\
\Leftrightarrow \quad & \lambda_k \sum_{i,j=1}^{l} \alpha_i^k \langle \phi(x_i), \phi(x_j) \rangle = \frac{1}{l} \sum_{i,j=1}^{l} \alpha_i^k \langle \phi(x_i), \phi(x_j) \rangle^2 \\
\Leftrightarrow \quad & \lambda_k \sum_{i,j=1}^{l} \alpha_i^k K(x_i, x_j) = \frac{1}{l} \sum_{i,j=1}^{l} \alpha_i^k K(x_i, x_j)^2 \\
\Leftrightarrow \quad & \underbrace{l \lambda_k}_{\eta_k} \boldsymbol{\alpha}^k = \mathbf{K} \boldsymbol{\alpha}^k
\end{aligned}
\tag{11}
$$

it can be seen that $\{\eta_k, \boldsymbol{\alpha}^k\}$ is actually an eigenvalue-eigenvector pair of matrix $\mathbf{K}$. In this way, the eigenvector decomposition of bilinear form C in $\mathcal{H}$ can be transformed to the decomposition of matrix $\mathbf{K}$.

Kernel PCA, cont.



(a) (b) (c) (d)

Figure: (a) A point cloud of table tennis racket; (b–d) reconstruction using the first 1–3 principal components. For each point in the bounding-box volume, the darkness is proportional to the density of the Gaussian in the feature space $\mathcal{H}$.

Un-centralized Case

$$\begin{aligned} \mathbf{u}^k &= \textstyle\sum_{i=1}^{l} \alpha_i^k \left(\phi(x_i) - \mu\right) \\ &= \textstyle\sum_{i=1}^{l} \alpha_i^k \left[\phi(x_i) - \frac{1}{l}\sum_{m=1}^{l} \phi(x_m)\right] \\ &= \phi(M)^\top \underbrace{\left(\mathbf{I}_l - \frac{1}{l}\mathbf{1}_l\mathbf{1}_l^\top\right)}_{\mathbf{I}^{\mathbf{E}}} \boldsymbol{\alpha}^k \end{aligned} \tag{12}$$

(where $\phi(M)^\top = [\phi(x_1), \phi(x_2), \cdots, \phi(x_l)]$, $\mathbf{1}_l$ is a l dimension vector with all entry equal 1, $\mathbf{I}_l$ is $l \times l$ identity matrix, $\boldsymbol{\alpha}^{k\top} = \left[\alpha_1^k, \alpha_2^k, \cdots, \alpha_l^k\right]$)

$$\mathbf{u}^{k\top}\mathbf{u}^h = 0, \quad \forall k \neq h \tag{13}$$

$$\mathbf{u}_1^k = \phi(M_1)^\top \mathbf{I}_1^{\mathbf{E}} \boldsymbol{\alpha}^k \tag{14}$$

$$\mathbf{u}_2^k = \phi(M_2)^\top \mathbf{I}_2^{\mathbf{E}} \boldsymbol{\alpha}^k \tag{15}$$

Rotation in Hilbert Space

Only **D** eigenvectors are used to represent the covariance of high dimension Gaussian distribution of each point cloud::

$$\mathbf{U}_1 = \left[\mathbf{u}_1^1, \cdots, \mathbf{u}_1^k, \cdots, \mathbf{u}_1^{\mathbf{D}}\right] \tag{16}$$

$$\mathbf{U}_2 = \left[\mathbf{u}_2^1, \cdots, \mathbf{u}_2^k, \cdots, \mathbf{u}_2^{\mathbf{D}}\right] \tag{17}$$

Align $\mathbf{U}_1$ with $\mathbf{U}_2$: $\mathbf{U}_2 = \mathbf{R}_{\mathcal{H}}\mathbf{U}_1$

$$\begin{aligned} & \mathbf{U}_2 = \mathbf{R}_{\mathcal{H}}\mathbf{U}_1 \\ \Longleftrightarrow \quad & \mathbf{U}_2\mathbf{U}_1^\top = \mathbf{R}_{\mathcal{H}}\mathbf{U}_1\mathbf{U}_1^\top \end{aligned} \tag{18}$$

$$\begin{aligned} \Longleftrightarrow \quad \mathbf{R}_{\mathcal{H}} &= \mathbf{U}_2\mathbf{U}_1^\top \\ &= \textstyle\sum_{k=1}^{\mathbf{D}} \mathbf{u}_2^k \mathbf{u}_1^{k\top} \\ &= \phi(M_2)^\top \underbrace{\mathbf{I}_2^{\mathbf{E}} \left(\sum_{k=1}^{\mathbf{D}} \alpha_2^k \alpha_1^{k\top}\right) \mathbf{I}_1^{\mathbf{E}}}_{\gamma} \phi(M_1) \end{aligned} \tag{19}$$

Translation in Hilbert Space

if $\mathbf{M}_1$ has already been centered, i.e. $\boldsymbol{\mu}_{\mathcal{H}}^{(1)} = 0$

$$\mathbf{b}_{\mathcal{H}} = \boldsymbol{\mu}_{\mathcal{H}}^{(2)} = \frac{1}{l_2}\phi(\mathbf{M}_2)^\top \mathbf{1}_{l_2} \tag{20}$$

3.Rigid Transformation in $\mathbb{R}^3$

Consistency

consistency error:

$$
\begin{tikzcd}
\mathbf{x}_t^{(1)} \arrow[r, "\phi"] \arrow[d, "\{\mathbf{R},\mathbf{b}\}"'] & \phi(\mathbf{x}_t^{(1)}) \arrow[d, "\{\mathbf{R}_\mathcal{H},\mathbf{b}_\mathcal{H}\}"] \\
\mathbf{R}\mathbf{x}_t^{(1)} + \mathbf{b} \arrow[r, "\phi"'] & \underbrace{\phi(\mathbf{R}\mathbf{x}_t^{(1)} + \mathbf{b})}_{\mathbf{\Phi}_t} \sim \underbrace{\mathbf{R}_\mathcal{H}\tilde{\phi}(\mathbf{x}_t^{(1)}) + \mathbf{b}_\mathcal{H}}_{\mathbf{\Psi}_t}
\end{tikzcd}
$$

$$\{\mathbf{R}^*, \mathbf{b}^*\} = \arg\min_{\mathbf{R},\mathbf{b}} \frac{1}{l_1} \sum_{t=1}^{l_1} \|\mathbf{\Psi}_t - \mathbf{\Phi}_t\|^2 \tag{21}$$

Because $\|\mathbf{\Phi}(\mathbf{x})\|^2$ can preserve constant under any translation $\mathbf{b}$ and rotation $\mathbf{R}$, and $\mathbf{\Psi}_t$ is fixed, :

$$\{\mathbf{R}^*, \mathbf{b}^*\} = \arg\max_{\mathbf{R},\mathbf{b}} \frac{1}{l_1} \sum_{t=1}^{l_1} \mathbf{\Phi}_t^\top \mathbf{\Psi}_t \tag{22}$$

Objective Function

$$\{\mathbf{R}^*, \mathbf{b}^*\} = \arg\max_{\mathbf{R},\mathbf{b}} \underbrace{\frac{1}{l_1}\sum_{t=1}^{l_1} \mathbf{\Phi}_t^\top \mathbf{\Psi}_t}_{\mathbf{O}} \tag{23}$$

$$\begin{aligned}
&\mathbf{O} \\
= \quad & \tfrac{1}{l_1}\textstyle\sum_{t=1}^{l_1} \left\{ \mathbf{\Phi}_t^\top \left[\underbrace{\phi(M_2)^\top \gamma \phi(M_1)}_{\mathbf{R}_{\mathcal{H}}} \left(\phi(x_t^{(1)}) - \underbrace{\frac{1}{l_1}\phi(M_1)^\top \mathbf{1}_{l_1}}_{\mu_1} \right) + \underbrace{\frac{1}{l_2}\phi(M_2)^\top \mathbf{1}_{l_2}}_{\mu_2} \right] \right. \\
= \quad & \tfrac{1}{l_1}\textstyle\sum_{t=1}^{l_1} K(\mathbf{R}x_t^{(1)} + \mathbf{b}, M_2)^\top \underbrace{\left[\gamma \left(K(x_t^{(1)}, M_1) - \frac{1}{k_1}\mathbf{K}_1 \mathbf{1}_{l_1} \right) + \frac{1}{l_2}\mathbf{1}_{l_2} \right]}_{\boldsymbol{\rho}_t} \\
= \quad & \tfrac{1}{l_1}\textstyle\sum_{t=1}^{l_1} \sum_{i=1}^{l_2} K(\mathbf{R}x_t^{(1)} + \mathbf{b}, x_i^{(2)}) \rho_{t,i}
\end{aligned} \tag{24}$$

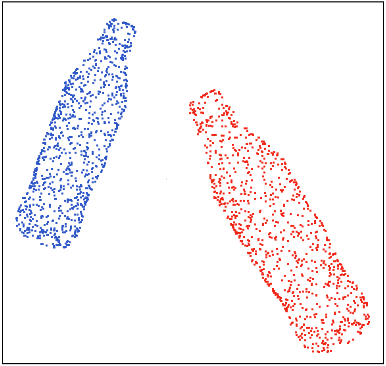
Simplified Objective Function

only a small number of points $\mathbf{D}+1$ is enough:

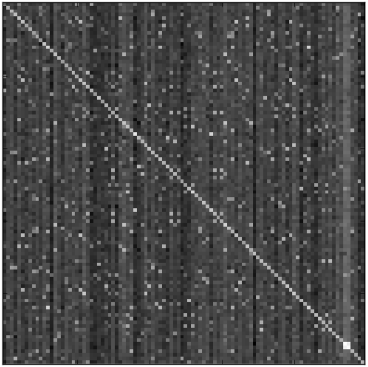
$$\{\mathbf{R}^*, \mathbf{b}^*\} = \arg\max_{\mathbf{R},\mathbf{b}} \frac{1}{\mathbf{D}+1} \frac{1}{l_2} \sum_{t=1,i=1}^{\mathbf{D}+1,l_2} K(\mathbf{R}x_{\mathbf{S}_t}^{(1)} + \mathbf{b}, x_i^{(2)})\rho_{t,i} \tag{25}$$

where $\mathbf{S}$ denotes the randomly selected subset of $\mathbf{M}_1$

Implicit Correspondence



(a)



(b)

Figure: (a) Two identical point clouds with exactly the same point permutation. (b) Visualization of ρ_t computed for all pairs of points.

Relation to Other Approaches

- our method:

$$\{\mathbf{R}^*, \mathbf{b}^*\} = \arg\max_{\mathbf{R},\mathbf{b}} \frac{1}{\mathbf{D}+1} \frac{1}{l_2} \sum_{t=1,i=1}^{\mathbf{D}+1,l_2} K(\mathbf{R}x^{(1)}_{\mathbf{S}_t} + \mathbf{b}, x^{(2)}_i)\rho_{t,i} \quad (26)$$

- SoftAssign /EM-ICP

$$\{\mathbf{R}^*, \mathbf{b}^*\} = \arg\min_{\mathbf{R},\mathbf{b}} \frac{1}{l_1} \sum_{t=1}^{l_1} \sum_{i=1}^{l_2} -\log K(\mathbf{R}x^{(1)}_t + \mathbf{b}, x^{(2)}_i) w_{t,i} \quad (27)$$

- Gaussian Mixtures

$$\{\mathbf{R}^*, \mathbf{b}^*\} = \arg\max_{\mathbf{R},\mathbf{b}} \frac{1}{l_1} \sum_{t=1}^{l_1} \sum_{i=1}^{l_2} K(\mathbf{R}x^{(1)}_t + \mathbf{b}, x^{(2)}_i) \quad (28)$$

Relation to Other Approaches, cont.

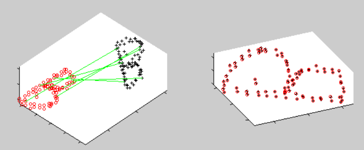
pseudo Gaussian mixture alignment:

$$\begin{aligned}
\mathbf{u}_1^k &= \phi(M_1)^\top \overbrace{\mathbf{I}_1^{\mathbf{E}} \boldsymbol{\alpha}^k}^{\boldsymbol{\beta}^k} \\
&= \textstyle\sum_{i=1}^{l_1} \beta_i^k \phi(x_i^{(1)}) \\
&= \textstyle\sum_{i=1}^{l_1} \tilde{\beta}_i^k \mathcal{N}(\xi; x_i^{(1)}, \sigma)
\end{aligned} \tag{29}$$

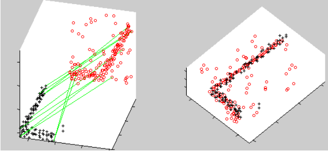
Remark:

- pseudo Gaussian mixture: $\tilde{\beta}_i^k$ can be negative;
- **D** pseudo Gaussian mixtures are aligned simultaneously.

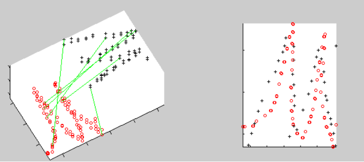
Qualitative Experiments



(a)



(b)



(c)

Figure: Test of the proposed algorithm in typical challenging circumstances for registration: (a) large motion; (b) outliers; (c) nonrigid transformation

Qualitative Experiments,cont.

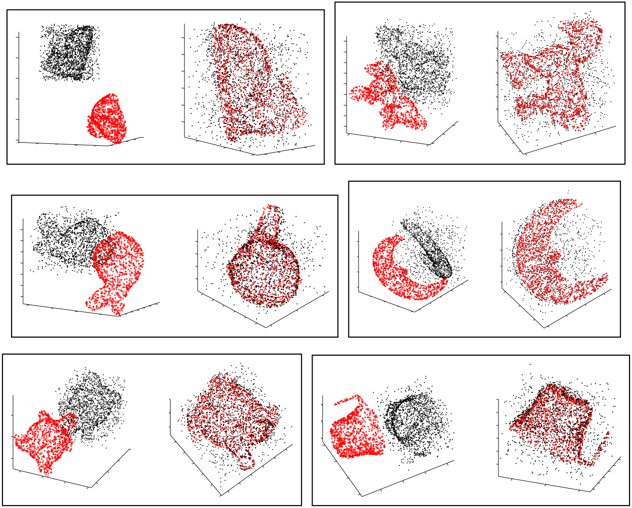


Figure: More test results on KIT 3D object database

Quantitative Experiments

Accuracy and Robustness

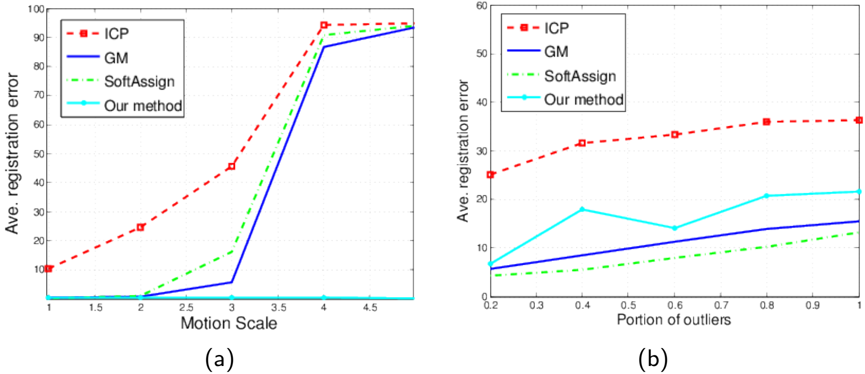


Figure: Test of four registration algorithm on (a) different scales of motions; (b) different portion of outliers added.

Quantitative Experiments, cont.

Efficiency

Point cloud size n	complexity	200	500	1000	2000
Our method	$n \log n$	1.489	2.162	5.126	21.165
ICP[BM92]	$n \log n$	0.023	0.051	0.154	0.469
GaussianMixtures[JV11]	n^2	3.998	15.245	43.570	172.4
SoftAssign[GRLM97]	n^2	4.801	83.925	592.1	3812

Table: Average execution time (seconds)

Conclusion

- kernel feature map point cloud to Hilbert space;
- align projections of point clouds in Hilbert space;
- project alignment back to $\mathbf{R}^3$;
- accurate and robust to large motion and outliers;
- much faster than state-of-the-art methods;

END
Questions are welcome !

P. J. Besl and H. D. Mckay.
A Method for Registration of 3-D Shapes.
PAMI, 14(2):239–256, 1992.

Steven Gold, Anand Rangarajan, Chienping Lu, and Eric Mjolsness.
New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence.
Pattern Recognition, 31:957–964, 1997.

Bing Jian and Baba C. Vemuri.
Robust Point Set Registration Using Gaussian Mixture Models.
PAMI, 33(8):1633–1645, 2011.

Computation Complexity Reduction

$$
\begin{aligned}
&\langle \mathbf{\Phi}_t, \mathbf{\Psi}_t \rangle \\
= \quad & \overline{\phi(\mathbf{P}\mathbf{x}_t^{(1)})}^\top \left(\sum_{k=1}^{\mathbf{D}} \tilde{\mathbf{u}}_2^k \tilde{\mathbf{u}}_1^{k\top} \left(\overline{\phi(\mathbf{x}_t^{(1)}) - \boldsymbol{\mu}_1} \right) + \boldsymbol{\mu}_2 \right) \\
= \quad & \sum_{k=1}^{\mathbf{D}} \langle \tilde{\mathbf{u}}_2^k, \overline{\phi(\mathbf{P}\mathbf{x}_t^{(1)})} \rangle \langle \tilde{\mathbf{u}}_1^k, \overline{\phi(\mathbf{x}_t^{(1)}) - \boldsymbol{\mu}_1} \rangle + \langle \boldsymbol{\mu}_2, \overline{\phi(\mathbf{P}\mathbf{x}_t^{(1)})} \rangle \qquad (30) \\
= \quad & \sum_{k=1}^{\mathbf{D}} \langle \tilde{\mathbf{u}}_2^k, \overline{\phi(\mathbf{P}\mathbf{x}_t^{(1)})} \rangle \langle \tilde{\mathbf{u}}_2^k, \mathbf{R}_{\mathcal{H}} \overline{\phi(\mathbf{x}_t^{(1)}) - \boldsymbol{\mu}_1} \rangle + \langle \boldsymbol{\mu}_2, \overline{\phi(\mathbf{P}\mathbf{x}_t^{(1)})} \rangle
\end{aligned}
$$

where we can see that $\overline{\phi(\mathbf{P}\mathbf{x}_t^{(1)})}$ and $\mathbf{R}_{\mathcal{H}} \overline{\phi(\mathbf{x}_t^{(1)}) - \boldsymbol{\mu}_1}$ are projected onto **D** eigenvectors $\left\{\tilde{\mathbf{u}}_2^k\right\}_{k=1}^{\mathbf{D}}$ respectively, and an additional projection of $\overline{\phi(\mathbf{P}\mathbf{x}_t^{(1)})}$ onto $\boldsymbol{\mu}_2$. Therefore, the computation of the objective function is actually done in a space spanned by **D** eigenvectors $\left\{\tilde{\mathbf{u}}_2^k\right\}_{k=1}^{\mathbf{D}}$ and one $\boldsymbol{\mu}_2$, which is a subspace of $\mathcal{H}$.