Three-Dimensional Sensors
Lecture 6: Point-Cloud Registration

Radu Horaud
INRIA Grenoble Rhone-Alpes, France
Radu.Horaud@inria.fr
http://perception.inrialpes.fr/

Point-Cloud Registration Methods

- "Fuse" data gathered with several 3D sensors or with a moving 3D sensor.
- Each sensor provides a point cloud in a sensor-centered coordinate frame.
- There is only a partial overlap between the two point clouds.
- The data are corrupted both by noise and by outliers (or gross errors).
- The two point clouds must be *registered* or represented in the same coordinate frame.
- The registration process requires point-to-point correspondences which is a difficult problem.

Estimating the Transformation Between Two Point Clouds

- We start by assuming that the two point clouds have an equal number of points and that the point-to-point correspondences are known in advance.
- Notations:
  - Each point cloud has N points $\{\boldsymbol{X}_n\}_{n=1}^{n=N}$ and $\{\boldsymbol{Y}_n\}_{n=1}^{n=N}$.
  - $\overline{\boldsymbol{X}}$ and $\overline{\boldsymbol{Y}}$ are the centers of the two point clouds.
  - $\widehat{\boldsymbol{X}}_n = \boldsymbol{X}_n - \overline{\boldsymbol{X}}$ and $\widehat{\boldsymbol{Y}}_n = \boldsymbol{Y}_n - \overline{\boldsymbol{Y}}$ are the centered coordinates.
  - The transformation between the two clouds is determined by a scale factor s, a rotation matrix $\mathbf{R}$ and a translation vector $\boldsymbol{t}$.
- A point in one cloud is the transformed of its corresponding point in the other cloud:

$$\boldsymbol{Y}_n = s\mathbf{R}\boldsymbol{X}_n + \boldsymbol{t}$$

Least-Square Minimization

- The transformation parameters can be determined by minimization of:

$$E = \sum_{n=1}^{N} \|\boldsymbol{Y}_n - s\mathbf{R}\boldsymbol{X}_n - \boldsymbol{t}\|^2$$

- Let's develop the quadratic error E (using $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$):

$$\begin{aligned} E &= (\boldsymbol{Y}_n - s\mathbf{R}\boldsymbol{X}_n - \boldsymbol{t})^\top(\boldsymbol{Y}_n - s\mathbf{R}\boldsymbol{X}_n - \boldsymbol{t}) \\ &= s^2\left(\sum_n \boldsymbol{X}_n^\top \boldsymbol{X}_n\right) + 2s\left(\sum_n \boldsymbol{X}_n^\top\right)\mathbf{R}^\top\boldsymbol{t} \\ &\quad -2s\left(\sum_n \boldsymbol{X}_n^\top \mathbf{R}^\top \boldsymbol{Y}_n\right) - 2\left(\sum_n \boldsymbol{Y}_n^\top\right)\boldsymbol{t} + \sum_n \boldsymbol{Y}_n^\top \boldsymbol{Y}_n + N\boldsymbol{t}^\top\boldsymbol{t} \end{aligned}$$

The Optimal Translation

- $\boldsymbol{t}^\star = \text{argmin}_{\boldsymbol{t}} E$
- The derivative of the error with respect to $\boldsymbol{t}$ writes:

$$\frac{dE}{d\boldsymbol{t}} = 2s\mathbf{R}\sum_n \boldsymbol{X}_n - 2\sum_n \boldsymbol{Y}_n + 2N\boldsymbol{t}$$

- Setting the derivative with respect to the translation vector equal to 0

$$\boldsymbol{t}^\star = \overline{\boldsymbol{Y}} - s\mathbf{R}\overline{\boldsymbol{X}}$$

A Simplified Criterion

- By substitution of the optimal translation in the error function, we obtain:

$$
\begin{aligned}
E &= \sum_{n=1}^{N} \|\widehat{\boldsymbol{Y}}_n - s\mathbf{R}\widehat{\boldsymbol{X}}_n\|^2 \\
&= s^2 \left(\sum_n \widehat{\boldsymbol{X}}_n^\top \widehat{\boldsymbol{X}}_n \right) - 2s \left(\sum_n \widehat{\boldsymbol{X}}_n^\top \mathbf{R}^\top \widehat{\boldsymbol{Y}}_n \right) + \sum_n \widehat{\boldsymbol{Y}}_n^\top \widehat{\boldsymbol{Y}}_n
\end{aligned}
$$

The Optimal Scale Factor

- $s^\star = \operatorname{argmin}_s E$
- Which yields:

$$s^\star = \frac{\sum_n (\mathbf{R}\widehat{\boldsymbol{X}}_n)^\top \widehat{\boldsymbol{Y}}_n}{\sum_n \widehat{\boldsymbol{X}}_n^\top \widehat{\boldsymbol{X}}_n}$$

- Using a *symmetric* treatment of the scale factor [Horn'87] shows that the error:

$$E = \sum_{n=1}^{N} \|(\sqrt{s})^{-1}\widehat{\boldsymbol{Y}}_n - \sqrt{s}\mathbf{R}\widehat{\boldsymbol{X}}_n\|^2$$

- yields the following solution:

$$s^\star = \frac{\sum_n \widehat{\boldsymbol{Y}}_n^\top \widehat{\boldsymbol{Y}}_n}{\sum_n \widehat{\boldsymbol{X}}_n^\top \widehat{\boldsymbol{X}}_n}$$

The Optimal Rotation

- From the expansion of the error function we obtain that:

$$\mathbf{R}^\star = \underset{\mathbf{R}}{\operatorname{argmin}}\, E = \underset{\mathbf{R}}{\operatorname{argmax}} \left(\sum_n \widehat{\boldsymbol{X}}_n^\top \mathbf{R}^\top \widehat{\boldsymbol{Y}}_n \right)$$

  and: $Q = \sum_n \widehat{\boldsymbol{Y}}_n^\top \mathbf{R} \widehat{\boldsymbol{X}}_n = \operatorname{tr}\left(\mathbf{R} \widehat{\boldsymbol{X}}_n \widehat{\boldsymbol{Y}}_n^\top \right) = \operatorname{tr}(\mathbf{R}\mathbf{C}_{XY})$

- Solution: Let $\mathbf{C}_{XY} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$ be the *singular value decomposition* (SVD) of the data co-variance matrix.
- The optimal rotation matrix is (see [ArunHuangBlostein'87] for more details):

$$\mathbf{R}^\star = \mathbf{V}\mathbf{U}^\top$$

Point Registration

- We assume now that the point-to-point correspondences are not known. The minimization criterion becomes:

$$E = \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_{mn} \| \boldsymbol{Y}_m - s\mathbf{R}\boldsymbol{X}_n - \boldsymbol{t} \|^2$$

- Possible representations for the *assignment variables* α_{mn}:
  1. **Binary** assignment variables: $\alpha_{mn} = 1$ if Y_m is in correspondence with X_n and $\alpha_{mn} = 0$ otherwise.
  2. **Soft** assignment variables: They are subject to the constraints $\sum_m \alpha_{mn} = \sum_n \alpha_{mn} = 1, 0 \leq \alpha_{mn} \leq 1$.
  3. **Probabilistic** assignment variables: $\alpha_{mn} = P(Y_m = n | Y_m)$ or the *posterior probability* of assigning point m to the point n.

The ICP Algorithm

- This is the most used point-registration method: *iterative closest point* (ICP) proposed by [BeslMcKay'92]:
  1. Initialize the scale, rotation and translation parameters;
  2. Transform each point X_n and search its closest point $Y_k = \text{argmin}_m \|\boldsymbol{Y}_m - s\mathbf{R}\boldsymbol{X}_n - \boldsymbol{t}\|^2$ and set $\alpha_{kn} = 1$;
  3. Estimate the parameters minimizing:

$$E = \sum_{k=1}^{M} \alpha_{nk} \|\boldsymbol{Y}_k - s\mathbf{R}\boldsymbol{X}_n - \boldsymbol{t}\|^2$$

  4. Repeat steps 2 and 3 until convergence (the parameters do not change any more).

Drawbacks of ICP

- It is very sensitive to the initial transformation parameters (it is trapped in a local solution).
- It is not robust to gross errors in the data.
- It cannot deal with anisotropic noise (the noise has the same variance along all the coordinates).

Accelerating ICP

- If the point clouds contain a large number of points, each iteration of the algorithm is rather slow. One can either sample the data, or better, apply K-means clustering to each point cloud. The cluster centers are then used in ICP.
- Step 2 is the most time-consuming process of the algorithm and needs an efficient implementation of searching for the nearest point. One can use KD-trees to structure one set of points as a binary tree and to rapidly perform a query into this tree in order to retrieve the closest point.
- Many variants of ICP were proposed in the vision/graphics literature.

Probabilistic Point Registration

- The minimization criterion is now generalized to:

$$E = \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_{mn} \left(\| \boldsymbol{Y}_m - s\mathbf{R}\boldsymbol{X}_n - \boldsymbol{t} \|^2_{\mathbf{C}_n} + \log |\mathbf{C}_n| \right)$$

- The posterior probabilities are defined by:

$$\alpha_{mn} = \frac{|\mathbf{C}_n|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \| \boldsymbol{Y}_m - s\mathbf{R}\boldsymbol{X}_n - \boldsymbol{t} \|^2_{\mathbf{C}_n} \right)}{\sum_{k=1}^{N} |\mathbf{C}_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \| \boldsymbol{Y}_m - s\mathbf{R}\boldsymbol{X}_k - \boldsymbol{t} \|^2_{\mathbf{C}_k} \right) + \emptyset_{3D}}$$

- Mahalanobis distance: $\| \boldsymbol{a} - \boldsymbol{b} \|^2_{\mathbf{C}} = (\boldsymbol{a} - \boldsymbol{b})^\top \mathbf{C}^{-1} (\boldsymbol{a} - \boldsymbol{b})$

Point Registration with Expectation-Maximization (EM)

1. Initialize the transformation parameters s, $\mathbf{R}$, $\boldsymbol{t}$ and the covariance matrices $\{\mathbf{C}_n\}_{n=1}^N$.
2. Estimate the posterior probabilities $\{\alpha_{mn}\}_{m=1,n=1}^{m=M,n=N}$.
3. Estimate the transformation parameters s, $\mathbf{R}$, $\boldsymbol{t}$.
4. Estimate the covariance matrices.
5. Repeat steps 2, 3, and 4 until convergence

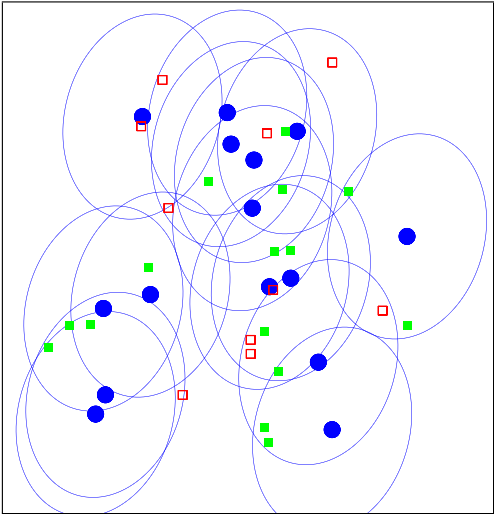
Discussion

- The EM algorithm for point registration uses a Gaussian mixture model (GMM) augmented with a uniform-distribution component that *captures* the outliers.
- The estimation of the transformation parameters do not need explicit point-to-point correspondences.
- However, such correspondences can be easily established using: if $\alpha_{mk} \geq 0.5$ accept the assignment $Y_m \leftrightarrow X_k$ with $\alpha_{mk} = \text{argmax}_n \, \alpha_{mn}$.
- The algorithm is robust to initialization, Gaussian noise and uniformly distributed gross errors (outliers), it may however be trapped in local minima.
- It is less efficient than ICP because of the computation of the probabilities α_{mn}.
- See [Horaud et al 2011] for more details.

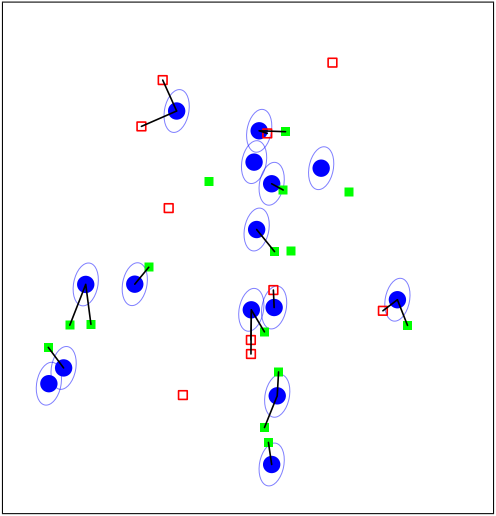
An Example

- The first set: 15 points (blue circles)
- The second set: 25 points of which 15 inliers corrupted with Gaussian noise (green squares) and 10 outliers drawn from a uniform distribution (red squares).
- Initialization: $\mathbf{R} = \mathbf{I}$ and $\boldsymbol{t} = 0$.
- Result: 12 point-to-point correspondences found correctly and 7 point-to-outlier assignments.

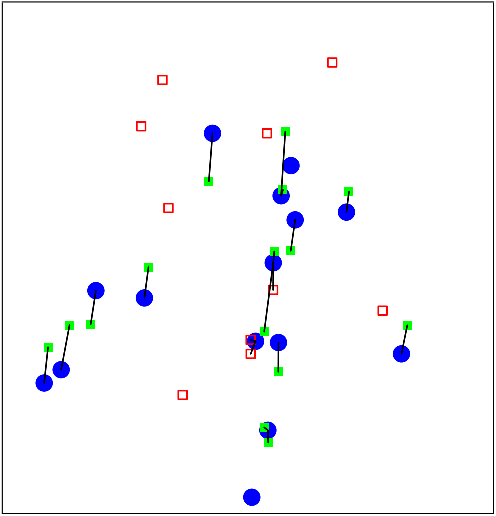
The 2-nd Iteration



The 6-th Iteration



The 35-th and Final Iteration



Projects Using 3-D Sensors and Point Registration

- The Great Budha project (University of Tokyo): http://www.cvl.iis.u-tokyo.ac.jp/greatbuddha.html
- KinectFusion project: http://research.microsoft.com/en-us/projects/surfacerecon/
- The MixCam project: http://perception.inrialpes.fr/MixCam/index.html