Zachary Paine Sabin

Seth Billings' IMLP: A Critical Review

Summary of Mathematical Approach

IMLP is an algorithm based on the same concepts as ICP. The goal is to "co-align multiple representations of a shape or environment" by calculating the transformation that aligns the representations. In ICP and IMLP, this is done with an iterative algorithm that first attempts to calculate the correspondences between points in the two representations or, if one of the representations is a mesh, calculate the correspondence between points in the cloud and points on the surface of the mesh. Once the algorithm has these correspondences it then uses them to calculate a transformation that aligns corresponding points as closely as possible using some distance measurement defined by the algorithm. ICP generally does this by creating correspondences between points in the source shape and the closest point to them in the target shape. Once it has these correspondences it then calculates a transformation such that the transformed source shape minimizes an error function, usually simply a sum of distances between corresponding points squared. This algorithm can be seen below (taken from Billings' paper).

Algorithm 1. Iterative Closest Point (ICP)

input: Source shape as point cloud: $X = \{\vec{x}_i\}$
Target shape: Ψ
Initial transformation estimate: $[R_0, \vec{t}_0]$
output: Final transformation $[R, \vec{t}]$ that aligns the shapes X and Ψ
1 Initialize transformation: $[R, \vec{t}] \leftarrow [R_0, \vec{t}_0]$
2 **while** *not converged* **do**
3 Compute closest-point correspondences $Y = \{\vec{y}_i\}$:
$\vec{y}_i \leftarrow C_{CP}(R\vec{x}_i + \vec{t}, \Psi)$
4 Update the transformation to minimize $E_{LS}(X, Y)$:

$$[R, \vec{t}] \leftarrow \underset{[R, \vec{t}]}{\operatorname{argmin}} \sum_{i=1}^{n} \| \vec{y}_i - R\vec{x}_i - \vec{t} \|_2^2$$

5 **end**

In his paper Billings proposes an alternative to ICP that he calls IMLP. The idea behind IMLP is that each point has anisotropic error associated with its position. He assumes that these errors are independent, zero-mean, multivariate Gaussians and thus calculates the likelihood of two points corresponding as being the equation below (Equation 3 from Billings' Paper).

$$L_{match}(\vec{x}, \vec{y}, M_x, M_y, R, \vec{t}) = \frac{1}{\sqrt{(2\pi)^3 \mid RM_xR^T + M_y \mid}} e^{-\frac{1}{2}(\vec{y} - R\vec{x} - \vec{t})^T (RM_xR^T + M_y)^{-1} (\vec{y} - R\vec{x} - \vec{t})} \quad (3)$$

In this equation, M_x and M_y are the covariance matrices for the error of the x and y points on the source and target shape respectively. Billings then uses this equation instead of simply the Euclidean

distance (as is done in ICP) to find the correspondences between points during the correspondence phase of his algorithm. In other words, rather than match points with the point closest to them on the target shape, he matches them with the point with the highest match likelihood. This equation forms the basis of IMLP, which can be seen in the below algorithm (from Billings' paper).

Algorithm 2. Iterative Most-Likely Point (IMLP)

input : Source shape as point cloud: $X = \{\vec{x}_i\}$
Target shape: Ψ
Measurement-error covariances: $M_X = \{M_{Xi}\}$, M_Ψ
Surface-model covariances: $M_{SX} = \{M_{Sxi}\}$, $M_{S\Psi}$
Initial transformation estimate: $[R_0, \vec{t}_0]$
Upper bound on match uncertainty: σ^2_{max} (default: ∞)
Chi-square threshold value for outliers: χ^2_{thresh} (default: 7.81)

output: Final transformation $[R, \vec{t}]$ that aligns the shapes X and Ψ

1 Initialize transformation: $[R, \vec{t}] \leftarrow [R_0, \vec{t}_0]$

2 Initialize noise model: $\sigma^2 \leftarrow 0$

3 Compute initial correspondences (Equ. 8):
$[\vec{y}_i, M_{yi}, M_{Syi}] \leftarrow C_{MLP}(\vec{x}_i, \Psi, I, I, R, \vec{t})$

4 Skip to Step 6

5 Compute most-likely correspondences (Equ. 8):
$[\vec{y}_i, M_{yi}, M_{Syi}] \leftarrow C_{MLP}(\vec{x}_i, \Psi, M_{xi} + M_{Sxi} + \sigma^2 I, M_\Psi + M_{S\Psi}, R, \vec{t})$

6 Update the match-uncertainty noise-model term (Equ. 4):
$\sigma^2 \leftarrow \min\left(\frac{1}{N_{inlier}} \sum_{i \in \text{ inliers}} \| \vec{y}_i - R\vec{x}_i - \vec{t} \|_2^2, \sigma^2_{max}\right)$

7 Identify outliers using a chi-square test (Equ. 6):
$(\vec{x}_i, \vec{y}_i)$ is outlier if $E_{SqrMahalDist}(\vec{x}_i, \vec{y}_i, M_{xi}, M_{yi} + \sigma^2 I, R, \vec{t}) > \chi^2_{thresh}$
and update the outlier noise-model terms (Equ. 7):

$$\varphi_i = \begin{cases} 9 \| \vec{y}_i - R\vec{x}_i - \vec{t} \|_2^2 & \text{if } (\vec{x}_i, \vec{y}_i) \text{ is an outlier} \\ 0 & \text{otherwise} \end{cases}$$

8 Set the noise-model covariances for the registration phase:
$M^*_{xi} \leftarrow M_{xi} + M_{Sxi} + \frac{\varphi_i}{2} I$, $M^*_{yi} \leftarrow M_{yi} + M_{Syi} + \frac{\varphi_i}{2} I + \sigma^2 I$

9 Update the transformation to align the corresponding point sets by GTLS (Equ. 20):

$$[R, \vec{t}] \leftarrow \underset{[R,\vec{t}]}{\text{argmin}} \sum_{i=1}^{n} (\vec{y}_i - R\vec{x}_i - \vec{t})^T (RM^*_{xi}R^T + M^*_{yi})^{-1} (\vec{y}_i - R\vec{x}_i - \vec{t})$$

10 **if** *not converged* **then** goto Step 5

While there is not enough space in this review to explore too deeply the math that Billings uses, this paper will quickly review the steps the algorithm takes before reviewing the separate sections of the paper.

In the inputs to the algorithm there are two covariance matrices for each point. The surface model covariances are directed along the surface of the target shape to "increase the noise-model variance in the surface-parallel directions" to encourage error to be along the surface of the target shape rather than normal to it. This is separated from the standard covariance as Billings claims that it was found to improve outlier rejection.

The match uncertainty term in the inputs to the algorithm is an isotropic variance that is added to the noise model and represents the "amount of misalignment between the source and target shapes." This is so that in the early stages of the algorithm, when the shapes are extremely misaligned, the covariance matrices are not supposed by the algorithm to account for the misalignment. This term is calculated at each step of the algorithm using the below equation (Equation 4 from Billings' paper).

$$\sigma^2 = \frac{1}{N_{\text{inlier}}} \sum_{i \in \text{inlier}} \| \vec{y}_i - R\vec{x}_i - \vec{t} \|_2^2$$

Outlier detection in IMLP is done using the Mahalanobis match distance, which defines the distance between two points as the number of standard deviations away the two points are along the distribution's principal component axes. This is defined in Equation 5 of Billings' paper.

$$\mathrm{E}_{\text{SqrMahalDist}}(\vec{x}, \vec{y}, M_x, M_y, R, \vec{t}) = (\vec{y} - R\vec{x} - \vec{t})^T (RM_xR^T + M_y)^{-1} (\vec{y} - R\vec{x} - \vec{t})$$

The algorithm sets any points whose distance exceeds the chi-squared threshold given as an input to the algorithm as outliers. The algorithm then adds isotropic noise relative to the distance between the corresponding points to the outliers' noise models, thus reducing their influence during the registration phase. This can be seen in Equation 7 of Billings' paper. Billings also points out that one could completely remove outliers from the registration phase, and he states that this is preferable in cases where the shapes have only partial overlap.

$$\varphi_i = \begin{cases} 9 \| \vec{y}_i - R\vec{x}_i - \vec{t} \|_2^2 & \text{if } (\vec{x}_i, \vec{y}_i) \text{ is an outlier} \\ 0 & \text{otherwise} \end{cases}$$

In order to find the correspondences between points, i.e. find the point on the target shape with the highest match likelihood for a point on the source shape, IMLP uses a PD tree. This is similar to a kd tree but each node defines a local coordinate system based on the eigenvectors of the covariance matrix. This review will not explore too deeply the workings of the PD tree, but Billings' search algorithm can be seen below.

Algorithm 3. PD-Tree Search for Most-Likely Correspondence

input : Source point: $\vec{x}$
Source-point noise model: M_x
PD tree containing target shape (Ψ) and target noise model (M_Ψ): T
Current transformation: $[R, \vec{t}]$
Prior most-likely match for this source point: $(\vec{y}_{\text{prev}}, M_{y\_\text{prev}})$
output: Most-likely match and its corresponding noise model: $(\vec{y}, M_y)$
1 Initialize most-likely match to the prior match:
$[\vec{y}, M_y, E_{\text{best}}] \leftarrow [\vec{y}_{\text{prev}}, M_{y\_\text{prev}}, \mathrm{E}_{\text{match}}(\vec{x}, \vec{y}_{\text{prev}}, M_x, M_{y\_\text{prev}}, R, \vec{t})]$
2 Search for more-likely match in the left child of the PD-tree root node:
$[\vec{y}_{\text{LChild}}, M_{y\_\text{LChild}}, E_{\text{LChild}}] \leftarrow \text{NodeSearch}(T.Root.LChild, E_{\text{best}}, \vec{x}, M_x, R, \vec{t})$
3 **if** $E_{LChild} < E_{best}$ **then** update most-likely match:
$[\vec{y}, M_y, E_{\text{best}}] \leftarrow [\vec{y}_{\text{LChild}}, M_{y\_\text{LChild}}, E_{\text{LChild}}]$
4 Search for more-likely match in the right child of the PD-tree root node:
$[\vec{y}_{\text{RChild}}, M_{y\_\text{RChild}}, E_{\text{RChild}}] \leftarrow \text{NodeSearch}(T.Root.RChild, E_{\text{best}}, \vec{x}, M_x, R, \vec{t})$
5 **if** $E_{RChild} < E_{best}$ **then** update most-likely match:
$[\vec{y}, M_y, E_{\text{best}}] \leftarrow [\vec{y}_{\text{RChild}}, M_{y\_\text{RChild}}, E_{\text{RChild}}]$

After explaining his use of the PD tree, Billings has a long section of his paper devoted to the different methods that one can use to bound the match error of a node (and thus bound the PD tree search). The algorithm has to check at least every point that satisfies the inequality below (Equation 10 from Billings' paper).

$$(\vec{y} - R\vec{x} - \vec{t})^T (RM_x R^T + M_y)^{-1} (\vec{y} - R\vec{x} - \vec{t}) < E_{\text{best}} - \log \mid RM_x R^T + M_y \mid .$$

He then presents a way to bound the log component of the inequality above, as well as three methods of bounding the square Mahalanobis match distance, the left side of the above inequality. This review will not explore these bounding methods but they can be found in Billings' paper.

Once Billings has established his method of finding the correspondences, he then discusses how to align the source and target shapes. He does this using least squares, similar to ICP, but instead of minimizing Euclidean distance he minimizes the total match error as defined in the below equation (19 from Billings' paper). However he claims that the Log term in the below equation can be disregarded as it is very small compared to the remainder of the equation, so the least squares problem is to minimize the square Mahalanobis-distance. He then explores how to solve this using least squares, which this review will not discuss.

$$\sum_{i=1}^{n} \log \mid RM_{xi} R^T + M_{yi} \mid + \sum_{i=1}^{n} (\vec{y}_i - R\vec{x}_i - \vec{t})^T (RM_{xi} R^T + M_{yi})^{-1} (\vec{y}_i - R\vec{x}_i - \vec{t})$$

Evaluation

Billings paper is a very thorough explanation of his mathematical justifications, algorithmic implementations, and experimental results. The Methods section discusses thoroughly why he made the choices that he did and that equations that were put into the algorithm. The strengths of this section were its thoroughness in exploring every equation and discussing alternatives in some cases. Billings also used experimental results to back up some of his choices (e.g. his choice to increase the variance of outliers rather than simply not include them altogether) which was a nice way to support his theory. However, some of the theoretical background of his equations was not completely explored which can leave the reader struggling to understand how Billings reached a certain equation. This may be the fault of the reader for not having a strong enough theoretical background but a couple extra sentences about how he reached certain results would make the paper far more readable. For example explaining the terms in his multivariate Gaussian distribution (i.e. transforming the covariance matrices into the target space and then adding them to calculate the net distribution) would be a nice touch.

His results and conclusion sections are also very thorough. He explains clearly each experiment and runs experiments on eight separate cases in which one might wish to run ICP or IMLP. He also tests the results of not just basic ICP and IMLP, but also of other "state of the art" algorithms such as robust ICP, Coherent Point Drift, and generalized ICP. He also attempts different variations of his algorithm, by switching most likely point to the closest point in Mahalanobis-distance and simply to closest point. Standard IMLP still outperforms almost all variations in terms of target registration error. However in

some cases, particularly those with high outlier rates, IMLP has significantly higher rates of registration failure (defined as a greater than 10 mm difference after registration) than other algorithms. Billings puts these results into difficult to read tables, whereas the target registration results are in graphs. It would have made for more readable results if he had put all results into graphical formats. He also does not discuss whether these registration failures are positive or negative, and it is not clear that they are negative (is it better to fail subtly or loudly?).

Overall his paper was thorough and established a strong theoretical backing and experimental results for his algorithm. It was also nice that he provided a fully coded implementation for the reader to look at.

Conclusion

IMLP is a strong algorithm and certainly appears to achieve significantly better results than standard ICP. While it does have a longer runtime this is not always important, especially as it is much more accurate in many cases. For our project, especially when we are calibrating the camera on the REMS robot, speed is not critical and there may be many outliers and lots of noise. Since IMLP handles outliers and noise with greater accuracy than ICP, we will certainly at least experiment with using it in our implementation.