ArticulatedFusion: Real-time Reconstruction of Motion, Geometry and Segmentation Using a Single Depth Camera —Supplemental Material—

Chao Li, Zheheng Zhao, and Xiaohu Guo

Department of Computer Science,
The University of Texas at Dallas
{Chao.Li2, Zheheng.Zhao, xguo}@utdallas.edu

1 Formula Derivation

In this section, we first introduce all symbols and then the derivation of related formula. The list of all symbols is shown in table 1:

Table 1. List of all symbols

C_k	the k-th cluster
$\mathbf{x}$	the position in canonical frame for a node of cluster C_k
$\mathbf{y}^t$	the position in live frame t for a node in cluster C_k
$\mathbf{c}_k$	the centroid position of cluster C_k in canonical frame
$\mathbf{c}_k^t$	the centroid position of cluster C_k in live frame t
n_k	the number of nodes belonging to cluster C_k in canonical frame
$\mathbf{A}^t(C_k)$	the cross covariance matrix of cluster C_k
$(R_k^t, \mathbf{t}_k^t)$	rotation and translation of cluster C_k
$(R^*, \mathbf{t}^*)$	optimal rotation and translation of cluster C_k
σ_{kq}	the q-th singular value of $\mathbf{A}^t(C_k)$

Optimal Energy E^* for an Arbitary Cluster

The total segmentation energy is as follows:

$$E_{seg} = \sum_{k=1}^{m} \sum_{\mathbf{x}\in C_k} \|R_k^t\mathbf{x} + \mathbf{t}_k^t - \mathbf{y}^t\|^2. \tag{1}$$

When segmentation is fixed, the segmentation energy term of each cluster is independent to each other. Therefore, we can parallelly compute the optimal energy of each cluster $E^*(C_k)$. In order to calculate $E^*(C_k)$, we first need to know how to compute the optimal value of rotation and translation $(R_n^t, \mathbf{t}_n^t)$ based on the segmentation energy term of an arbitary cluster C_k:

$$E(C_k) = \sum_{\mathbf{x} \in C_k} \|R_k^t \mathbf{x} + \mathbf{t}_k^t - \mathbf{y}^t\|^2. \tag{2}$$

According to Sorkine-Hornung and Rabinovich's technical report [1], the optimal $(R_n^t, \mathbf{t}_n^t)$ is as follows:

$$R^* = V \begin{pmatrix} 1 & & \\ & 1 & \\ & & det(VU^\top) \end{pmatrix} U^\top, \tag{3}$$

$$\mathbf{t}^* = \mathbf{c}_k^t - R^* \mathbf{c}_k, \tag{4}$$

where

$$\mathbf{A}^t(C_k) = PQ^\top = U\Sigma V^\top, \tag{5}$$

$$P = \begin{pmatrix} \cdots \\ \mathbf{x} - \mathbf{c}_k \\ \cdots \end{pmatrix}, \tag{6}$$

$$Q = \begin{pmatrix} \cdots \\ \mathbf{y}^t - \mathbf{c}_k^t \\ \cdots \end{pmatrix}, \tag{7}$$

$$PQ^\top = \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \mathbf{c}_k)(\mathbf{y}^t - \mathbf{c}_k^t)^\top, \tag{8}$$

$$\mathbf{c}_k = \frac{\sum_{\mathbf{x} \in C_k} \mathbf{x}}{n_k}, \tag{9}$$

$$\mathbf{c}_k^t = \frac{\sum_{\mathbf{y}^t \in C_k} \mathbf{y}^t}{n_k}. \tag{10}$$

$U\Sigma V^\top$ is the Singular Value Decomposition (SVD) of matrix $\mathbf{A}^t(C_k)$. Replace $\mathbf{t}_k^t$ in Eq.(2) by $\mathbf{t}^*$ in Eq.(4), we have:

$$\begin{aligned} E^*(C_k) &= \sum_{\mathbf{x} \in C_k} \|R^* \mathbf{x} + \mathbf{t}^* - \mathbf{y}^t\|^2 \\ &= \sum_{\mathbf{x} \in C_k} \|R^*(\mathbf{x} - \mathbf{c}_k) - (\mathbf{y}^t - \mathbf{c}_k^t)\|^2 \\ &= \sum_{\mathbf{x} \in C_k} [(\mathbf{x} - \mathbf{c}_k)^\top (\mathbf{x} - \mathbf{c}_k) + (\mathbf{y}^t - \mathbf{c}_k^t)^\top (\mathbf{y}^t - \mathbf{c}_k^t)] \\ &\quad - 2 \sum_{\mathbf{x} \in C_k} (\mathbf{y}^t - \mathbf{c}_k^t)^\top R^* (\mathbf{x} - \mathbf{c}_k). \end{aligned} \tag{11}$$

By making $\mathbf{p} := \mathbf{x} - \mathbf{c}_k$ and $\mathbf{q} := \mathbf{y}^t - \mathbf{c}_k^t$, according to Sorkine-Hornung and Rabinovich's technical report [1], we have:

$$\begin{aligned} &\sum_{\mathbf{x} \in C_k} (\mathbf{y}^t - \mathbf{c}_k^t)^\top R^* (\mathbf{x} - \mathbf{c}_k) \\ &= \sum_{\mathbf{x} \in C_k} \mathbf{q}^\top R^* \mathbf{p} \\ &= tr(Q^\top R^* P) = tr(R^* P Q^\top) = tr(R^* U \Sigma V^\top) \\ &= tr(\Sigma V^\top R^* U). \end{aligned} \tag{12}$$

According to Eq.(3), Eq.(12) can be re-written as:

$$\sum_{\mathbf{x} \in C_k} (\mathbf{y}^t - \mathbf{c}_k^t)^\top R^* (\mathbf{x} - \mathbf{c}_k) = \sum_{q=1}^{3} \sigma_{kq}. \tag{13}$$

Therefore,

$$E^*(C_k) = \sum_{\mathbf{x} \in C_k} [(\mathbf{x} - \mathbf{c}_k)^\top (\mathbf{x} - \mathbf{c}_k) + (\mathbf{y}^t - \mathbf{c}_k^t)^\top (\mathbf{y}^t - \mathbf{c}_k^t)] - 2 \sum_{q=1}^{3} \sigma_{kq}. \tag{14}$$

Cluster Mergence: $(C_i, C_j) \rightarrow C_k$

By decomposing Eq.(14) into two parts, we have

$$E_1^*(C_k) = \sum_{\mathbf{x} \in C_k} [(\mathbf{x} - \mathbf{c}_k)^\top (\mathbf{x} - \mathbf{c}_k) + (\mathbf{y}^t - \mathbf{c}_k^t)^\top (\mathbf{y}^t - \mathbf{c}_k^t)], \tag{15}$$

$$E_2^*(C_k) = -2 \sum_{q=1}^{3} \sigma_{kq}. \tag{16}$$

If we merge a pair of clusters (C_i, C_j) into a new cluster C_k, $E_1^*(C_k)$ and $E_2^*(C_k)$ can be updated in constant time.

Firstly,

$$\begin{aligned} \mathbf{A}^t(C_k) &= \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \mathbf{c}_k)(\mathbf{y}^t - \mathbf{c}_k^t)^\top \\ &= \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{c}_k)(\mathbf{y}^t - \mathbf{c}_k^t)^\top + \sum_{\mathbf{x} \in C_j} (\mathbf{x} - \mathbf{c}_k)(\mathbf{y}^t - \mathbf{c}_k^t)^\top \\ &= n_i(\mathbf{c}_i - \mathbf{c}_k)(\mathbf{c}_i^t - \mathbf{c}_k^t)^\top + n_j(\mathbf{c}_j - \mathbf{c}_k)(\mathbf{c}_j^t - \mathbf{c}_k^t)^\top \\ &\quad + \mathbf{A}^t(C_i) + \mathbf{A}^t(C_j), \end{aligned} \tag{17}$$

where n_i is the number of nodes in cluster C_i, and so is n_j for C_j. The new centroid $\mathbf{c}_k$ of cluster C_k can be updated by:

$$\mathbf{c}_k = \frac{n_i \mathbf{c}_i + n_j \mathbf{c}_j}{n_i + n_j}. \tag{18}$$

So is $\mathbf{c}_k^t$. And σ_{kq} can be solved by the SVD decomposition of $A^t(C_k)$. Therefore, $E_2^*(C_k)$ can be updated in constant time.

Secondly,

$$\begin{aligned} E_1^*(C_k) =& E_1^*(C_i) + E_1^*(C_j) \\ &+ n_i(\mathbf{c}_i - \mathbf{c}_k)^\top(\mathbf{c}_i - \mathbf{c}_k) + n_i(\mathbf{c}_i^t - \mathbf{c}_k^t)^\top(\mathbf{c}_i^t - \mathbf{c}_k^t) \\ &+ n_j(\mathbf{c}_j - \mathbf{c}_k)^\top(\mathbf{c}_j - \mathbf{c}_k) + n_j(\mathbf{c}_j^t - \mathbf{c}_k^t)^\top(\mathbf{c}_j^t - \mathbf{c}_k^t). \end{aligned} \tag{19}$$

Therefore, $E_1^*(C_k)$ can be updated in constant time.

Because both $E_1^*(C_k)$ and $E_2^*(C_k)$ can be updated in constant time, the energy change aftering merging C_i, C_j into C_k can be calculated in constant time as follows:

$$\begin{aligned} \Delta E_1^* =& E_1^*(C_k) - E_1^*(C_i) - E_1^*(C_j) \\ =& n_i(\mathbf{c}_i - \mathbf{c}_k)^\top(\mathbf{c}_i - \mathbf{c}_k) + n_i(\mathbf{c}_i^t - \mathbf{c}_k^t)^\top(\mathbf{c}_i^t - \mathbf{c}_k^t) \\ &+ n_j(\mathbf{c}_j - \mathbf{c}_k)^\top(\mathbf{c}_j - \mathbf{c}_k) + n_j(\mathbf{c}_j^t - \mathbf{c}_k^t)^\top(\mathbf{c}_j^t - \mathbf{c}_k^t), \end{aligned} \tag{20}$$

$$\Delta E_2^* = E_2^*(C_k) - E_2^*(C_i) - E_2^*(C_j), \tag{21}$$

$$\Delta E^* = \Delta E_1^* + \Delta E_2^*. \tag{22}$$

In summary, the merge cost ΔE^* can be computed efficiently by keeping track of the centroid, the number of nodes and the cross covariance matrix of each cluster: $\{\mathbf{c}_k, n_k, \mathbf{A}^t(C_k)\}$, simple computing operations in constant time and a SVD decomposition of a 3×3 cross covariance matrix.

Cluster Optimization: Swapping C_l from C_i to C_j

Let us suppose swapping a one-node cluster C_l from C_i to C_j (C_l only contains one boundary node $\mathbf{x}_l$). After swapping, C_i becomes $C_{i'}$ and C_j becomes $C_{j'}$: i.e., $C_{i'} = C_i - C_l$, $C_{j'} = C_j \cup C_l$. Then we can consider that C_i is formed by a merging operation $(C_{i'}, C_l) \to C_i$ and $C_{j'}$ is formed by a merging operation $(C_j, C_l) \to C_{j'}$. This can help us treat the swapping operation as a combination of merging operations, and calculate energy change by using equations from the previous section.

The decrease of energy E_2^* is:

$$\Delta E_2^* = E_2^*(C_{i'}) + E_2^*(C_{j'}) - E_2^*(C_i) - E_2^*(C_j). \tag{23}$$

$E_2^*(C_{i'}) + E_2^*(C_{j'})$ can be calculated by SVD of $\mathbf{A}^t(C_{i'})$ and $\mathbf{A}^t(C_{j'})$. $\mathbf{A}^t(C_{i'})$ and $\mathbf{A}^t(C_{j'})$ can be updated by proper substitution according to Eq. (17):

$$\begin{aligned} \mathbf{A}^t(C_{i'}) =& \mathbf{A}^t(C_i) - \mathbf{A}^t(C_l) \\ &- n_{i'}(\mathbf{c}_{i'} - \mathbf{c}_i)(\mathbf{c}_{i'}^t - \mathbf{c}_i^t)^\top - (\mathbf{x}_l - \mathbf{c}_i)(\mathbf{y}_l^t - \mathbf{c}_i^t)^\top, \end{aligned} \tag{24}$$

$$\begin{aligned}\mathbf{A}^t(C_{j'}) =&\mathbf{A}^t(C_j) + \mathbf{A}^t(C_l)\\ &+ n_j(\mathbf{c}_j - \mathbf{c}_{j'})(\mathbf{c}_j^t - \mathbf{c}_{j'}^t)^\top + (\mathbf{x}_l - \mathbf{c}_{j'})(\mathbf{y}_l^t - \mathbf{c}_{j'}^t)^\top.\end{aligned} \tag{25}$$

Then following the same rules in calculation of ΔE_2^*, the decrease of energy E_1^* is:

$$\begin{aligned}\Delta E_1^* =&E_1^*(C_{i'}) + E_1^*(C_{j'}) - E_1^*(C_i) - E_1^*(C_j)\\ =&n_j(\mathbf{c}_j - \mathbf{c}_{j'})^\top(\mathbf{c}_j - \mathbf{c}_{j'}) + n_j(\mathbf{c}_j^t - \mathbf{c}_{j'}^t)^\top(\mathbf{c}_j^t - \mathbf{c}_{j'}^t)\\ &+ (\mathbf{x}_l - \mathbf{c}_{j'})^\top(\mathbf{x}_l - \mathbf{c}_{j'}) + (\mathbf{y}_l - \mathbf{c}_{j'})^\top(\mathbf{y}_l^t - \mathbf{c}_{j'}^t)\\ &- n_{i'}(\mathbf{c}_{i'} - \mathbf{c}_i)^\top(\mathbf{c}_{i'} - \mathbf{c}_i) - n_{i'}(\mathbf{c}_{i'}^t - \mathbf{c}_i^t)^\top(\mathbf{c}_{i'}^t - \mathbf{c}_i^t)\\ &- (\mathbf{x}_l - \mathbf{c}_i)^\top(\mathbf{x}_l - \mathbf{c}_i) - (\mathbf{y}_l^t - \mathbf{c}_i^t)^\top(\mathbf{y}_l^t - \mathbf{c}_i^t).\end{aligned} \tag{26}$$

In summary, the swapping cost can be computed efficiently by keeping track of the centroid, the number of nodes, and the cross covariance matrix of each cluster: $\{\mathbf{c}_k, n_k, \mathbf{A}^t(C_k)\}$, simple computing operations in costant time and SVD decompositions of two 3×3 cross covariance matrices.

2 Discussion on Dynamic Clustering

Influence of Different Number of Clusters on Results

Suppose n is the necessary number of clusters for a motion, m is the total number of nodes, and k is the fixed number of clusters to choose. There are 3 situations:

1. $k < n$: i.e., insufficient number of clusters. The reconstruction result will be bad (Fig. 1 (a)). The highlighted right arm part is not well reconstructed.
2. $k \geq n$ and $k \ll m$: the reconstruction result will be good (Fig. 1 (b) and (c)).
3. Otherwise, the situation becomes similar as "DT + DynamicFusion". The reconstruction result will be still bad.

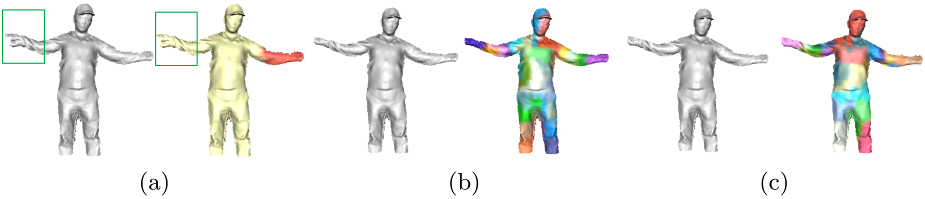
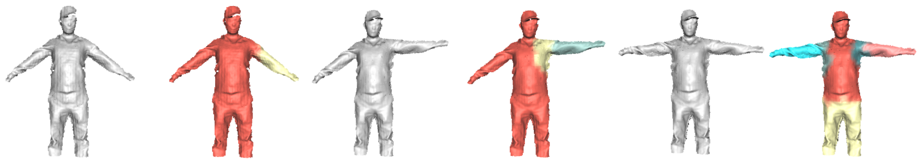


Fig. 1. (a) Reconstruction result with fixed 2 clusters. (b) Reconstruction result with fixed 30 clusters. (c) Reconstruction result with dynamic clustering. For each group of subfigures: the left-hand side image is the reconstructed geometry and the right-hand side image is current segmentation.

Dynamic clustering mechanism will automatically determine the number of clusters by setting an energy threshold in the merging step to decide when the merging should be stopped. In this way, the number of clusters is related to the complexity of motions. As Fig. 2 shows, when the motion is changed from one-arm-raising to two-arm-raising, the number of clusters is increased from 2 to 6 to represent the complex motion.



(a) (b) (c)

Fig. 2. Dynamic Clustering mechanism. From (a) to (b): number of clusters is changed from 2 to 3 when one arm is raising. From (b) to (c): number of clusters is changed from 3 to 6 when two arms are raising. For each group of subfigures: the left-hand side image is the reconstructed geometry and the right-hand side image is current segmentation.

References

1. Sorkine-Hornung, O., Rabinovich, M.: Least-squares rigid motion using svd. Department of Computer Science, ETH Zurich (2016)