Supplementary Material for "KinectAvatar: Fully Automatic Body Capture Using a Single Kinect"

1 Conformal Geometric Algebra

In this section, we give a brief introduction of the Conformal Geometric Algebra, and a relationship and framework transform between Conformal Geometric Algebra and the normal Euclidean Algebra.

In Conformal Geometric Algebra, we use the motor M to express the rotation and translation. The main question is now, how to solve a set of constraint equations for multiple features with respect to the unknown motor M. Since a motor is a polynomial of infinite degree, this is a non-trivial task, especially in the case of real-time estimation. How to get a linear equation with respect to the generators of the motor? We try to solve this problem with exponential representation of motors and the Taylor series expansion with the first approximation order. This leads to a mapping of the above mentioned global motion transformation to a twist representation, which allows for incremental changes of pose.

In this section, we derive the linearization of the motor. This results in linear equations in the generators of the unknown 3D rigid body motion. For simplicity, we consider the case of point transformations. The Euclidean transformations of a point $\underline{X}$ in conformal space caused by the motor M is approximated as:

$$\begin{aligned} M\underline{X}\tilde{M} &= \exp\left(-\frac{\theta}{2}\left(l' + e_\infty m'\right)\right)\underline{X}\exp\left(\frac{\theta}{2}\left(l' + e_\infty m'\right)\right) \\ &\approx \left(1 - \frac{\theta}{2}\left(l' + e_\infty m'\right)\right)\underline{X}\left(1 + \frac{\theta}{2}\left(l' + e_\infty m'\right)\right) \\ &\approx E + e_\infty\left(x - \theta(l' \cdot x) - \theta m'\right) \end{aligned} \tag{1}$$

We assume $l := \theta l'$ and $m := \theta m'$, then:

$$M\underline{X}\tilde{M} \approx E + e_\infty\left(x - l \cdot x - m\right) \tag{2}$$

Based on the definition above, we can use two vectors $l \in R3$ (l_1, l_2, l_3) and $m \in R3$ (m_1, m_2, m_3) express motor M. The transformation for point $x \in R3$ (x_1, x_2, x_3) can be expressed in Euclidean Algebra:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \rightarrow \begin{pmatrix} m_1 + x_1 - x_2 l_3 + x_3 l_2 \\ m_2 + x_2 - x_3 l_1 + x_1 l_3 \\ m_3 + x_3 - x_1 l_2 + x_2 l_1 \end{pmatrix} \tag{3}$$

Due to the approximation $\approx$ in equation (1), the unknown motion parameters l and m are linear. This equation contains six unknown parameters ((l_1, l_2, l_3) and (m_1, m_2, m_3)) for the rigid body motion.

The linear equations are solved for a set of correspondences by applying the Householder method. From the solution of the system of equations, the motion parameters R,

t can easily be recovered by evaluating $\theta := \|l\|, l' := \frac{l}{\theta}, m' := \frac{m}{\theta}$. The motor M can be evaluated by applying the Rodrigues' formula.

The principle of this approximation is illustrated in Fig. 1. The aim is to rotate a point $\underline{X}$ by 90 degrees to a point $\underline{X}'$. The first order approximation of the rotation leads to the tangent of the circle passing through X. Normalizing the tangent line to $\underline{X}'$(denoted by dashed lines) $\underline{X}_1$ is gained as the first order approximation of the required point $\underline{X}'$. By repeating this procedure the points $\underline{X}_2 \ldots \underline{X}_n$ will be estimated, approaching to the point $\underline{X}'$. It is clear from figure 1 that the convergence rate of a rotation depends on the amount of the expected rotation. All angles converge during the iteration. For the most cases just a few iterations are sufficient to get a good approximation. In situations where only small rotations are assumed, four iterations are sufficient for all cases.

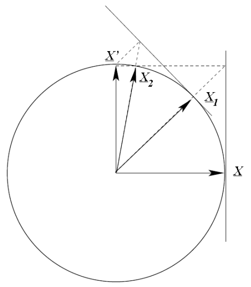

Fig. 1. Principle of convergence for the iteration of a point X rotated around 90 degrees to a point X'. X_1 is the result of the first iteration and X_2 is the result of the second iteration.

2 Mathematical Details of the Registration Algorithm

In this section, we provide additional mathematical details to the registration algorithm. In particular, we discuss the computations of the E-step and M-step in more detail.

E-step. For each pair of frames f and g, we need to update $P^{\text{old}}_{f,g}(m|y_{f,n})$ which is a matrix of dimension $N_g \times N_f$. The values of this matrix determine which correspondences $y_{f,n},\ y_{g,m}$ will be included within the error term during the M-step.

We transform each point $y_{f,n}$ using the rigid transformation assigned to that point $\mathcal{T}_f^{i(n)}$ (the transformation of index $i(n)$ in frame f). We express the transformation in the conformal geometric algebra as follows:

$$\mathcal{T}_f^{i(n)}(y_{f,n}) = M_f^{i(n)} y_{f,n} \tilde{M}_f^{i(n)}$$

Similarly, we transform $y_{g,m}$ using the rigid transformation $\mathcal{T}_g^{i(m)}$. The complete procedure for determining the matrix values for all pairs:

- For all frames $f = 1 \dots K$
  - For all frames g such that $f - 4 \leq g \leq f + 4$ (within four frames of f)
    - For all $n = 1 \dots N_f$
      - Find points in frame g that yield the smallest value according to the following formula:

        $$||\mathcal{T}_f^{i(n)}\left(y_{f,n}\right) - \mathcal{T}_g^{i(m)}\left(y_{g,m}\right)||^2$$

        To do this, we first transform each point $y_{g,m}$ of frame g using its transformation $\mathcal{T}_g^{i(m)}$. Rebuild the k-d tree for nearest neighbor search. For each point, we measure the distance to $\mathcal{T}_f^{i(n)}\left(y_{f,n}\right)$ and find the N_{near} points that are closest. We store these points in a set denoted "Near($y_{f,n}$)."
    - At this point, we have a set Near($y_{f,n}$) of closest points for each $y_{f,n}$. We compute the variance of the pair of frames f, g by averaging the squared distance to all closest points as follows:

      $$\sigma_{f,g}^2 = \frac{1}{N_f N_{\text{near}}} \sum_{n=1}^{N_f} \sum_{m \in \text{Near}(y_{f,n})} \left\| \mathcal{T}_f^{i(n)}\left(y_{f,n}\right) - \mathcal{T}_g^{j(m)}\left(y_{g,m}\right) \right\|^2$$
    - For all $n = 1 \dots N_f$
      - For all points $m \in \text{Near}(y_{f,n})$, compute the entry $p_{m,n}$ in the posterior matrix $P_{f,g}^{\text{old}}(m|y_{f,n})$ using the following formula:

        $$p_{m,n} = \frac{\exp\left(\frac{||\mathcal{T}_f^{i(n)}(y_{f,n}) - \mathcal{T}_g^{i(m)}(y_{g,m})||^2}{2\sigma_{f,g}^2}\right)}{\sum\limits_{m \in \text{Near}(y_{f,n})} \exp\left(\frac{||\mathcal{T}_f^{i(n)}(y_{f,n}) - \mathcal{T}_g^{i(m)}(y_{g,m})||^2}{2\sigma_{f,g}^2}\right)}$$

        All other entries $p_{m,n} = 0$. The denominator in the above formula has the effect of normalizing each column, so that the sum of each column equals 1.

M-step. Based on the E-step values and corresponding frames, the new alignment parameter values are found by minimizing the negative log-likelihood function, or more specifically, its upper bound Q which evaluates to:

$$Q\left(\mathcal{M}, \mathcal{L}\right) = \sum_{f,g} \left(\sum_{n=1}^{N_f} \sum_{m=1}^{N_g} P_{f,g}^{\text{old}}\left(m \mid y_{f,n}\right) \frac{\left\| \mathcal{T}_f^{i(n)}\left(y_{f,n}\right) - \mathcal{T}_g^{j(m)}\left(y_{g,m}\right) \right\|^2}{2{\sigma_{f,g}}^2} \right)$$

Since the deformation parameters change after each M-step, the variances are recomputed after the M-step update. We perform the M-step in two sub-steps iteratively until convergence.

Sub-Step 1. Fix labels $\mathcal{L}$ and solve for transformations $\mathcal{M}$. For E_{data}, the labels $i(n)$ and $j(m)$ are fixed, and the variables are the transformations M. For the regularization

E_{reg}, only the joint constraint remains, since the labels are fixed. We use the same optimization method as the rigid registration, except that the joint constraints are added as additional terms, and we solve for more transformations simultaneously.

Sub-Step 2. Fix transformations $\mathcal{M}$ and solve for labels $\mathcal{L}$. The labels $i(n)$ are the variables that we are solving for, and this affects the location of the points because it changes which transformation is being applied. Therefore, the goal is to re-segment the points to yield a better registration. In E_{reg}, the joint constraint can be ignored, and only the label constraint is left to ensure that the number of segmented parts in each frame is not too high. We solve the resulting discrete optimization problem using the α-expansion algorithm.

Effect of N_{near}. The parameter N_{near} is a fixed value for all experiments. We chose the value of $N_{\text{near}} = 20$ to balance between registration quality and the running time. Larger values of N_{near} helps the algorithm to be robust to noisy data. However, in our experiments, we discovered that a very high value does not necessary improve the registration result. For data from the Kinect, a value of N_{near} from 15 to 30 gives a good registration quality without sacrificing too much performance.