Autonomous Tracking For Volumetric Video Sequences: Supplementary Material

Matthew Moynihan
V-SENSE
Trinity College Dublin
mamoynih@tcd.ie

Susana Ruano
V-SENSE
Trinity College Dublin
ruanosas@tcd.ie

Rafael Pagés
Volograms
rafa@volograms.com

Aljosa Smolic
V-SENSE
Trinity College Dublin
smolica@tcd.ie

1. Overview

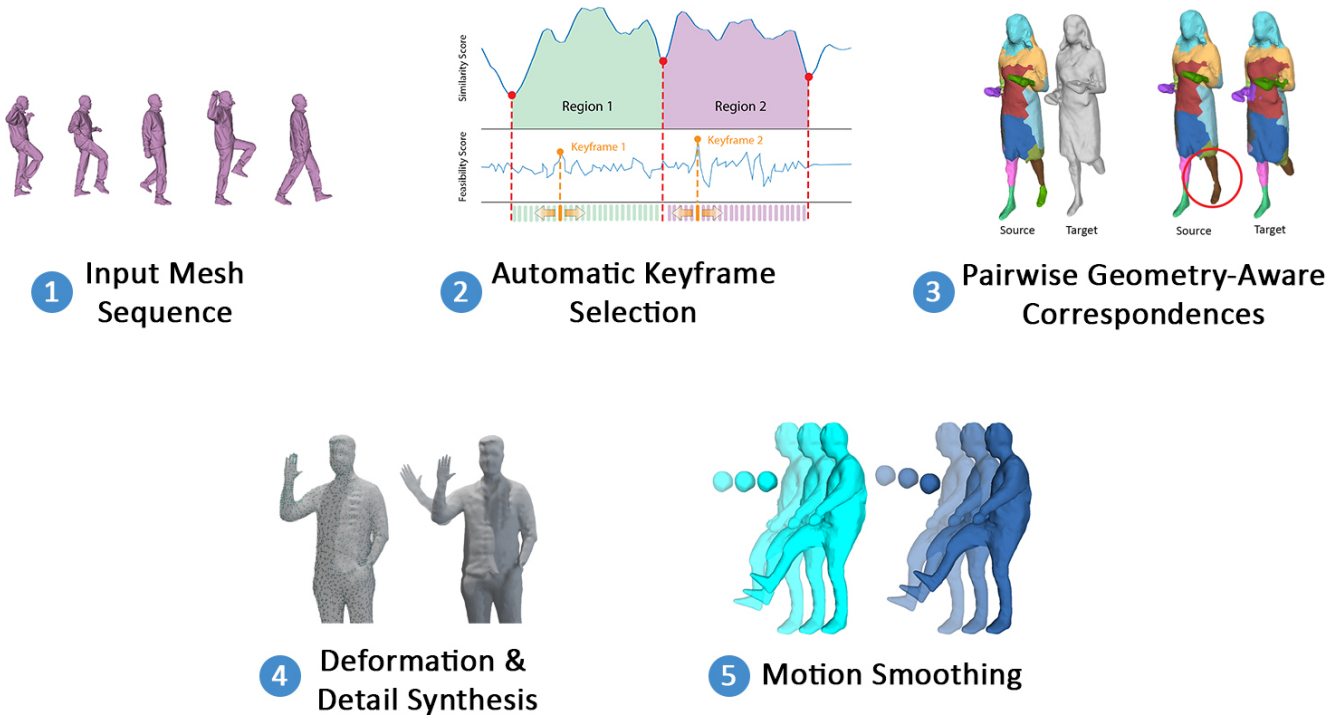

Figure 1. 1. The input is an incoherent sequence of meshes with independent topology. 2. Using shape similarity and abstraction proxies, a combination of similarity and feasibility scores are used to select keyframes which approximate the optimum selection of frames that will lead to successful pairwise registration across the sequence. 3. Between pairwise registrations, correspondences are found using volumetric segmentation and geometry-aware correspondences which support the recovery of missing geometry and allows for user-defined editing. 4. Using the correspondences, a deformation graph deforms the source mesh to the target. Detail synthesis is then performed to recover high-frequency details and reduce keyframe ”popping” effects. 5. 3D motion smoothing is applied to further improve the temporal coherence of the output tracked mesh sequence.

2. Implementation Details

The proposed system was implemented in C++ on Ubuntu 18.04.4 LTS on a laptop with an Intel i7-9750H CPU. On average, our system solves for 20k vertices per mesh in 45s per sequential alignment in comparison to 60s for [3] and 30s for [1]. While the proposed system does not outperform regarding speed, it is still competitive while producing significantly better results as demonstrated in our experiments and video.

2.1. Keyframing

We modify the feasibility score of Collet et. al [2] as the following:

$$S_i = \sum_{c \in C(i)} \left(1 + 2 * (g_{max} - g_c) + \frac{A_c}{2 * (A_{max} + 1)}\right) * \lambda_i \tag{1}$$

Where S_i is the feasibility score for frame i, C is the number of connected components and λ_i is the boundary weight which discourages keyrames at region edges and is formulated similarly to an activation function where x is the number of frames from the region boundary:

$$\lambda_i = 1 - \frac{1}{1 + x^2} \tag{2}$$

Empirically we found that surface area has less impact on the effect of tracking for noisy input data, especially considering that the proposed algorithm is designed to accommodate missing geometry. Genus has a significantly large impact on the appearance of "chewing gum" stretching artifacts which are still a major concern.

Guo et. al [3] use an L0-regularization to determine anchor frames, however, attempts to replicate this on real studio data failed as their system requires a template mesh to initialize as well as relatively noise-free target meshes for tracking. For these reasons we were unable to provide a similar analysis against their approach.

2.2. Correspondence Conditioning and Alignment

In this section we provide extra implementation details about the correspondence estimation process. For segmentation transfer we initialize the process with global rigid alignment followed by a two-way ICP match with normal-constrained alignment. In the event that some vertices in the target mesh do not have a segment match with the moving mesh we use a k-connected (k=2) region with a majority voting system to assign values to the highly sparse, unmatched vertices.

We also provide an extended description of the key data terms in the deformation graph equation. Equation 3, describes the main minimization equation as in Guo et. al [3]

$$E_{total} = E_{data} + \alpha_{rigid} E_{rigid} + \alpha_{smooth} E_{smooth} \tag{3}$$

Where E_{data} is the data term which expands to describe the point-to-point and point-to-plane correspondence error for a vertex v_j which has a matching vertex in the set of correspondences C:

$$E_{data} = \sum_{v_j \epsilon C} \alpha_{point} \left\| v_j^{'} - c_j \right\|_2^2 + \alpha_{plane} \left| n_{c_j}^T \left(v_j^{'} - c_j \right) \right|^2 \tag{4}$$

The E_{rigid} term encourages as-rigid-as-possible deformation and is constructed as:

$$\begin{aligned} E_{rigid} = \sum_j ((a_{j1}^T a_{j2})^2 + (a_{j2}^T a_{j3})^2 + (a_{j3}^T a_{j1})^2 + \\ (1 - a_{j1}^T a_{j1})^2 + (1 - a_{j2}^T a_{j2})^2 + (1 - a_{j3}^T a_{j3})^2) \end{aligned} \tag{5}$$

Where a_{j1}, a_{j2}, a_{j3} are the column vectors of R_j. The final term, E_{smooth} penalises abrupt variance between adjacent nodes and is given by:

$$E_{smooth} = \sum_{n_j} \sum_{n_i \epsilon N_{(n_j)}} w(n_i, n_j) \left\| R_i(n_i - n_j) + n_j + t_j - (n_i + t_i) \right\|_2^2 \tag{6}$$

This is formulated in our Gauss-Newton solver which is built using the Eigen[1] libraries and CHOLMOD[2] for Supernodal Sparse Cholesky Factorization and converges in less than 5 iterations under the criteria that $\Delta E_{total} < 1e-6$ between successive iterations. We use $\approx$ 2.5K nodes and $\approx$ 3K constraints on each deformation. We use $\alpha_{rigid} = 500$, $\alpha_{smooth} = 500$, $\alpha_{point} = 0.1$ and $\alpha_{plane} = 1.0$ similar to those values as recommended by Guo et. al [3].

2.3. Geometry Recovery

Missing geometry is not only flagged to be excluded by pointwise correspondence matching, but also we ignore the velocity-dependant smoothing for recovered geometry and instead opt for a static covariance noise in order to allow for motion interpolation of recovered data as is seen in Figure 8 (right) of the main paper.

2.4. Detail Synthesis

We extended the description of detail synthesis in the text with figure 2 which provides a more visual guide to the process.

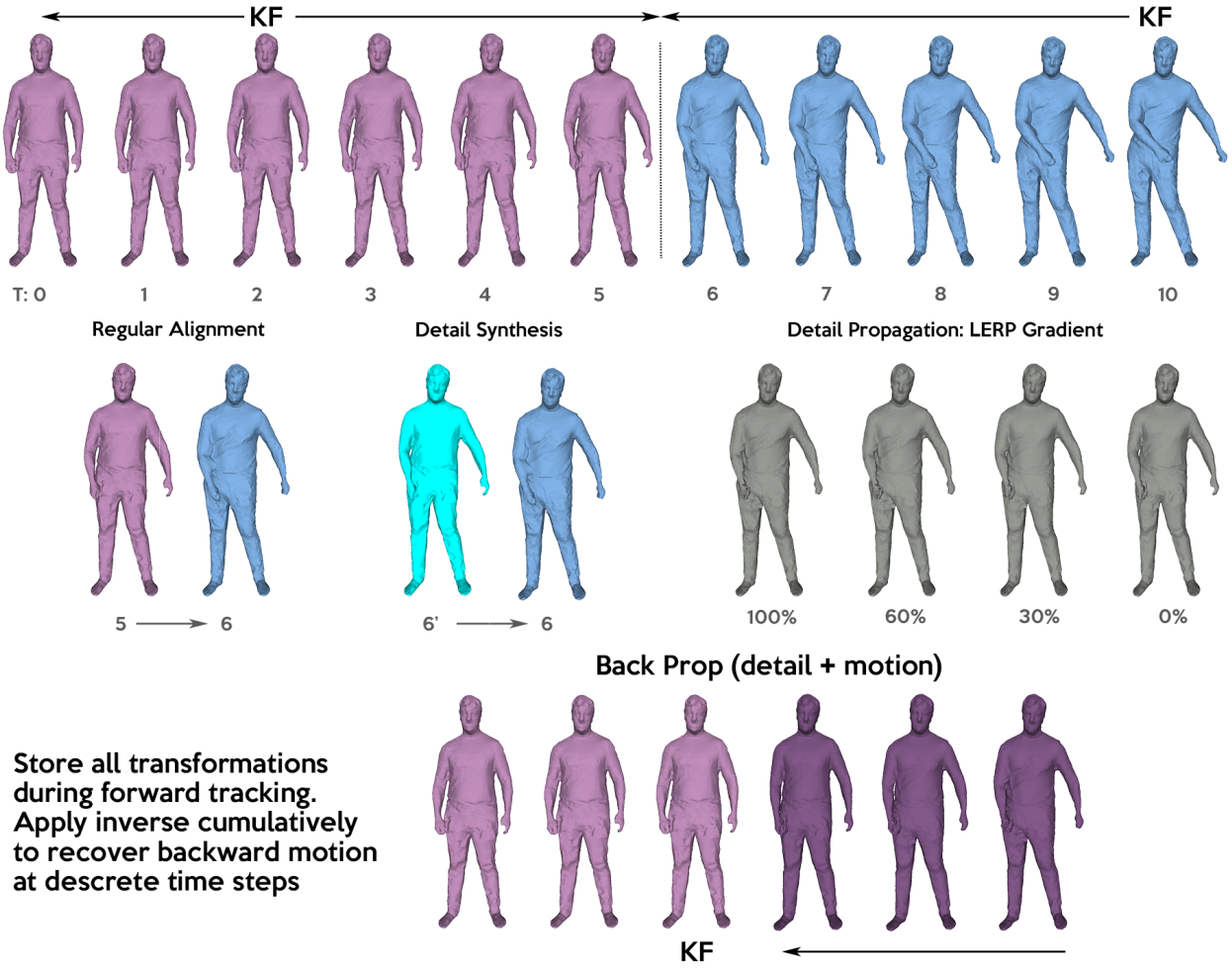

Figure 2. Detail synthesis. Given two regions pictured top where T is a global frame index. We have keyframes at $T = 2$ and $T = 10$, from which tracking is performed to 5 and 6 respectively. For resolving details from $2 \longrightarrow 5$ we track 5 towards 6 as a normal framewise alignment giving $6'$. We then track $6'$ to 6 which greatly relaxed rigidity parameters in order to synthesis surface details. This detail layer is then linearly interpolated (LERP) for n intervals between 0% and 100% where n is the number of frames between the keyframe and the boundary. We finally use the cached transformations from the original tracking to propagate the LERP intervals to their respective frames i.e. in the above example 5(100%), 4(60%) etc... We apply the same process for detail synthesis from $6 \longrightarrow 10$.

[1] http://eigen.tuxfamily.org/

[2] https://developer.nvidia.com/cholmod

References

[1] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[2] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015.

[3] Kaiwen Guo, Feng Xu, Yangang Wang, Yebin Liu, and Qionghai Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3083–3091, 2015.