Non-Rigid Structure from Locally-Rigid Motion

Supplemental Material

Jonathan Taylor Allan D. Jepson Kiriakos N. Kutulakos
Department of Computer Science
University of Toronto
{jtaylor,jepson,kyros}@cs.toronto.edu

A Derivation of the Projected-Length Equation

Squaring both sides of Eq. (5) in the paper we obtain

$$\left(\sqrt{L_{21}-l_{21}} \pm \sqrt{L_{32}-l_{32}}\right)^2 = L_{13}-l_{13} . \tag{12}$$

After expanding the left-hand side of Eq. (12) and rearranging terms, the expression becomes

$$(L_{21}-l_{21})+(L_{32}-l_{32})-(L_{13}-l_{13}) = \pm 2\sqrt{L_{21}-l_{21}}\,\sqrt{L_{32}-l_{32}} . \tag{13}$$

Squaring both sides of Eq. (13) eliminates the remaining square roots and sign ambiguities, leaving us with a quadratic expression in terms of squared distances in the image and in 3D:

$$\Big[(L_{21}-l_{21})+(L_{32}-l_{32})-(L_{13}-l_{13})\Big]^2 = 4(L_{21}-l_{21})(L_{32}-l_{32}) \tag{14}$$

The Projected-Length Equation now follows by expanding the squares in Eq. (14) and expressing the resulting quadratic polynomial as an inner product.

B Edge Weight and Pairwise Alignment Functionals (Section 4.2)

Let r_n^a be the binary variable representing the reflection state of triangle a in image n. Each variable is a node in the graph defined in Section 4.2. Recall that this graph contains two types of edges, encoding the following two constraints:

- *hinge-constraint edges*: the relative angle between hinge edges in a flexible pair should be the smallest possible in every image; and
- *pose-constraint edges*: the pose of each triangle changes as little as possible from one image to the next.

Therefore,

- if triangles a and b are a flexible pair, there is an edge between r_n^a and r_n^b for all $n = 1, \ldots, N$; and
- there is an edge between r_n^a and r_{n+1}^a for every triangle a.

Below we define the weights associated with these edges along with the associated pairwise alignment costs. We used these functionals, unaltered, for all our reconstructions.

B.1 Edge Weight Functional

Since there are two types of edges, we assign weights differently for each type. For hinge constraint edges we use

$$\mathcal{W}(r_n^a, r_n^b) \;=\; \begin{cases} 0.5 + \left(1 - \frac{|\theta^{\text{EQ}} - \theta^{\text{OPP}}|}{180}\right) \frac{\min(\theta^{\text{EQ}},\, \theta^{\text{OPP}})}{90} & |\theta^{\text{EQ}} - \theta^{\text{OPP}}| > 30 \text{ and} \\ & \min(\theta^{\text{EQ}},\ \theta^{\text{OPP}}) < 10 \\ \infty & \text{otherwise} \end{cases} \tag{15}$$

where a and b are two triangles in a flexible pair; θ^{EQ} is the angle between their hinge edges (in degrees) when $r_n^a = r_n^b$ and θ^{OPP} is their angle when $r_n^a \neq r_n^b$. This assigns a positive weight between triangles whose hinge edges are sufficiently well aligned in one of their two states (*i.e.,* within 10 degrees) and whose hinge edge is not too close to being fronto-parallel (*i.e.,* where flipping their reflection state does not cause a misalignment more than 30 degrees). For flexible pairs that meet these conditions, their weight in Eq. (15) implements a "soft" version of the conditions, assigning higher weight to pairs that approach their limits.

For pose constraint edges, we use

$$\mathcal{W}(r_n^a, r_{n+1}^a) \;=\; 1 - \frac{|\phi^{\text{EQ}} - \phi^{\text{OPP}}|}{180} \tag{16}$$

where ϕ^{EQ} is the angle between normals of triangle a in images n and $n+1$ when $r_n^a = r_{n+1}^a$ and ϕ^{EQ} is their angle when $r_n^a \neq r_{n+1}^a$. The less a triangle's pose has changed between the two images, the lower this weight will be. This makes it easier to propagate a triangle's reflection state from one image in the sequence to the next. Note that minimum possible weight in Eq. (16) is less than that in Eq. (15). This gives "preference" to constraints propagating within a single triangle in the temporal dimension as opposed to those propagating across triangles in the same instant. In practice, we found that adding this bias toward temporal propagation made an appreciable difference in the reflection state assignment results.

B.2 Pairwise Alignment Cost Functional

The pairwise alignment cost functional is used to decide which reflection state to assign to a child given the state of its parent. Again, the function depends on whether the edge is from a hinge constraint or a pose constraint. For hinge constraints we use

$$\mathcal{C}(r_n^a = r_n^b) \;=\; \frac{|\theta^{\text{EQ}}|}{90} \quad \text{and} \quad \mathcal{C}(r_n^a \neq r_n^b) \;=\; \frac{|\theta^{\text{OPP}}|}{90}\;, \tag{17}$$

where angles were defined as above. This essentially chooses for the child the reflection state that causes the least misalignment for the hinge edges. Temporal alignment costs are defined in an identical manner:

$$\mathcal{C}(r_n^a = r_{n+1}^a) \;=\; \frac{|\phi^{\text{EQ}}|}{90} \quad \text{and} \quad \mathcal{C}(r_n^a \neq r_{n+1}^a) \;=\; \frac{|\phi^{\text{OPP}}|}{90}\;. \tag{18}$$

C Description of Videos

Our videos are encoded using the MPEG4-XVID codec. If you have trouble viewing them (or if they appear in an incorrect orientation) we recommend using the cross-platform VLC Player (`http://www.videolan.org/vlc/`).

1. *1-wind-video.avi:* Reconstruction results for the *wind* sequence, where the input was created by projecting mocap data. *Top left:* Input images (i.e. feature positions). *Top right:* Triangles identified as rigid, along with their object membership. *Bottom left:* View of reconstruction from the camera's viewpoint. *Bottom right:* Side view of the reconstruction. Red spheres and the yellow wireframe show the vertices and edges of the triangles, respectively, reconstructed by locally-rigid SFM. As ground truth is available for the mocap sequences, blue dots show the 3D marker positions of the mocap data. Since our reconstruction is quite accurate, the red and blue spheres overlap for most of the video and thus only one of them is usually visible.

2. *2-jacky-video.avi:* Reconstruction results for images created from a second mocap dataset. We follow the same conventions as above.

3. *3-scarf-video.avi:* Results for the *scarf* sequence in Figure 6 (second last row). We follow the same conventions as above.

4. *4-person-video.avi:* Results for the person sequence in [22]. We follow the same conventions as above. We only show the reconstruction for the torso since the other groups had too few triangles to get a visual sense of the reconstructed geometry. Note the slight downward movement of the arm and elbow, causing a deformation of the torso, toward the end of the video. Triangles were grouped into two bodies (red and blue) because the tearing action caused all other in-between triangles to be identified as non-rigid. Although all red triangles were grouped together, the graph used for assigning reflection states had more than one connected components (see last paragraph of Section 4.2). The largest component component of this body is marked with red and the remaining triangles are shown in light red.

5. *5-two_cloths-video.avi:* Results for the *two-cloth* sequence in Figure 4 (right). We follow the same conventions as above.

6. *6-paper-video.avi:* Results for the sequence in [37] (also shown in Figure 6, third last row). *Far Left:* Triangle soup before determining reflection states. *Middle Left:* Triangle soup with assigned reflection states. *Middle Right:* Full reconstruction after depths are determined. *Far Right:* Image sequence rendered onto the image plane.

7. *7-tear-video.avi:* Results on the *tear* sequence from Figure 6 (last row).

8. *8-cars4-video.avi:* Results on a sequence from [19] that was not discussed in the paper due to lack of space. *Left:* Input images and tracked features. *Right:* Triangles identified as rigid, with color indicating their group. For this sequence, we only show segmentation results—since each group contained very few triangles, and since we compute no information about the relative 3D positions of different triangle groups, the reconstructions themselves were not informative.

9. *8-cars4-video.avi:* Segmentation results on another sequence from [19]. We follow the same conventions as above.

D Summary of Parameters

The table below summarizes our algorithm's five parameters. With the exception of the first one, ϵ, they were kept fixed in all experiments.

Parameter	**Value**	**Reference**
Reprojection error threshold (ϵ^*)	0.4 or 0.8	Section 3.1
Minimum 3D angle between two edges in a triangle	10 degrees	Section 3.1
Maximum 3D length of a triangle edge	$2.5 \times$ median 3D edge length across all triangles	Section 3.1
Minimum hinge-edge misalignment	10 degrees	Eq. (15)
Minimum hinge-edge alignment deviation	30 degrees	Eq. (15)