Inspired Quadrangulation - Appendix

Julien Tierny [1] Joel Daniels II [2] Luis G. Nonato [3] Valerio Pascucci [2] Claudio T. Silva [2,4]

Appendix A. 2D ARAP MLS transformation gradient

We provide an analytic expression of the gradient of as-rigid-as-possible planar maps, for a given constraint q_k. A closed-form formula for planar similarities is provided in [1]. For some given constraints p_i and their mapping $q_i = \widehat{\psi}_S(p_i)$, for each $x \in \mathcal{D}$ with (u, v) coordinate, the similarity transformation, noted $\widehat{\psi}_S$, is given by:

$$\widehat{\psi}_S(x) = q_* + \frac{\sum_i w_i \; \hat{q}_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i{}^{\perp} \end{pmatrix} \begin{pmatrix} x - p_* \\ -(x - p_*)^{\perp} \end{pmatrix}^T}{\mu_S} \tag{A.1}$$

where $w_i = \frac{1}{(p_i - x)^2}$, $p_* = \frac{\sum_i w_i p_i}{\sum_i w_i}$, $q_* = \frac{\sum_i w_i q_i}{\sum_i w_i}$, $\hat{p}_i = p_i - p_*$ and $\hat{q}_i = q_i - q_*$, ^{T} denotes the transpose and $(u, v)^{\perp} = (-v, u)$ and $\mu_S = \sum_i w_i \hat{p}_i \hat{p}_i{}^T$

A theorem is provided in [1], noticing that if locally the similarity can be re-written as a rotation matrix, it minimizes the energy functional of rigid transformations (see theorem 2.1). Then, the closed-form formula for planar ARAP transformations $\widehat{\psi}_R$ is the same as Eq. A.1, except that μ_S is switched for μ_R:

$$\mu_R = \sqrt{\left(\sum_i w_i \hat{q}_i \hat{p}_i{}^T\right)^2 + \left(\sum_i w_i \hat{q}_i \hat{p}_i{}^{\perp T}\right)^2} \tag{A.2}$$

For clarity, we derive the first component (u) of $\widehat{\psi}_R$, wrt u *(i)*, then wrt v *(ii)*. The derivation of the second component (noted v) of the gradient is obtained similarly and will be omitted. $\widehat{\psi}_R$ can be re-written in the form:

$$\widehat{\psi}_R(x) = q_* + \frac{\sum_i \hat{q}_i A_i}{\mu_R} \tag{A.3}$$

where $A_i = w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i{}^{\perp} \end{pmatrix} \begin{pmatrix} x - p_* \\ -(x - p_*)^{\perp} \end{pmatrix}^T = \begin{pmatrix} a_0^i & a_2^i \\ a_1^i & a_3^i \end{pmatrix}$ is an expression independent of q_i which can be pre-computed. Then, the u-component of $\nabla^k \left(\widehat{\psi}_R(x)\right)_u$, noted $\nabla_u^k \left(\widehat{\psi}_R(x)\right)_u$ is given by:

$$\boxed{\nabla_u^k \left(\widehat{\psi}_R(x)\right)_u = \nabla_u^k q_{u*} + \frac{\mu_R \; \nabla_u^k \left(\sum_i \hat{q}_i A_i\right)_u - \left(\sum_i \hat{q}_i A_i\right)_u \; \nabla_u^k \mu_R}{\mu_R^2}} \tag{A.4}$$

Notice μ_R is a scalar and $()_u$ denotes the u-component of a vector. The derivation of q_u* is straightforward (an infinitely small variation in $(q_k)_u$ will be multiplied by $\frac{w_k}{\sum_i w_i}$ while the v component remains unchanged):

$$\nabla^k q_{u*} = \left(\frac{w_k}{\sum_i w_i} \;\; 0\right) \tag{A.5}$$

$$\left(\sum_i \hat{q}_i A_i\right)_u = \sum_i \left((\hat{q}_i)_u \; a_0^i + (\hat{q}_i)_v \; a_1^i\right) \tag{A.6}$$

$$\nabla_u^k \left(\sum_i \hat{q}_i A_i\right)_u = a_0^k - w_k \sum_i \frac{a_0^i}{\sum_i w_i} \tag{A.7}$$

If we re-write μ_R as $\mu_R = \sqrt{(\mu_0^2 + \mu_1^2)}$, then we have:

$$\nabla_u^k \mu_R = \frac{1}{2}(\mu_0^2 + \mu_1^2)^{-\frac{1}{2}} \left(2\mu_0 \nabla_u^k \mu_0 + 2\mu_1 \nabla_u^k \mu_1\right) \tag{A.8}$$

$$\mu_0 = \sum_i \left((\hat{q}_i)_u (\hat{p}_i)_u + (\hat{q}_i)_v (\hat{p}_i)_v\right) \tag{A.9}$$

$$\mu_1 = \sum_i \left((\hat{q}_i)_u (\hat{p}_i{}^{\perp})_u + (\hat{q}_i)_v (\hat{p}_i{}^{\perp})_v\right) \tag{A.10}$$

$$\nabla_u^k \mu_0 = w_k (\hat{p}_k)_u - w_k \sum_i \frac{w_i (\hat{p}_i)_u}{\sum_i w_i} \tag{A.11}$$

$$\nabla_u^k \mu_1 = w_k (\hat{p}_k{}^{\perp})_u - w_k \sum_i \frac{w_i (\hat{p}_i{}^{\perp})_u}{\sum_i w_i} \tag{A.12}$$

Email addresses: `tierny@telecom-paristech.fr` (Julien Tierny), `jdaniels@sci.utah.edu` (Joel Daniels II), `gnonato@icmc.usp.br` (Luis G. Nonato), `pascucci@sci.utah.edu` (Valerio Pascucci), `csilva@nyu.edu` (Claudio T. Silva).

[1] CNRS - LTCI - Telecom ParisTech

[2] SCI Institute - University of Utah

[3] ICMC - Universidade de Sao Paulo

[4] NYU-Poly

Thus *(i)*, $\nabla_u^k \left(\widehat{\psi}_R(x)\right)_u$ (Eq. A.4) can be computed by combining the equations A.5, A.7, A.11 and A.12. Now, we need to derive again the u component of $\widehat{\psi}_R(x)$, but wrt v:

$$\boxed{\begin{aligned}\nabla_v^k \left(\widehat{\psi}_R(x)\right)_u &= \nabla_v^k q_{u*} \\ &+ \frac{\mu_R \ \nabla_v^k \left(\sum_i \hat{q}_i A_i\right)_u - \left(\sum_i \hat{q}_i A_i\right)_u \ \nabla_v^k \mu_R}{\mu_R^2}\end{aligned} \quad (A.13)}$$

$$\nabla_v^k \left(\sum_i \hat{q}_i A_i\right)_u = a_1^k - w_k \sum_i \frac{a_1^i}{\sum_i w_i} \quad (A.14)$$

$$\nabla_v^k \mu_0 = w_k (\hat{p_k})_v - w_k \sum_i \frac{w_i (\hat{p}_i)_v}{\sum_i w_i} \quad (A.15)$$

$$\nabla_v^k \mu_1 = w_k (\hat{p_k}^\perp)_v - w_k \sum_i \frac{w_i (\hat{p_i}^\perp)_v}{\sum_i w_i} \quad (A.16)$$

Then (*ii*), $\nabla_v^k \left(\widehat{\psi}_R(x)\right)_u$ can be computed by combining the equations A.5, A.14, A.15 and A.16 into the equation A.13. Finally, to complete the gradient computation, the v component of $\widehat{\psi}_R$ needs to be derived twice in the same manner: *(i)* with a small variation of q_k in u, *(ii)* with a small variation of q_k in v. We do not detail this derivation since it is highly similar to the derivations detailed above (a noticeable difference is that a_0^i and a_1^i need respectively to be switched for a_2^i and a_3^i, while the derivations of μ_0 and μ_1 do not change).

The computation of $\nabla\widehat{\psi}_R$ is very efficient. The following terms are computed *offline* (before the optimization), as soon as the p_i set is known: w_i, $\sum_i w_i$, A_i, $\hat{p}_i$ and $\hat{p_i}^\perp$. The following terms are computed during the *online* reconstruction of $\widehat{\psi}_R$: q_i, μ_R, μ_0 and μ_1. Thus, for a given constraint q_k, the gradient computation algorithm only needs to directly evaluate the right hand side of equations A.5, A.7, A.11, A.12, A.14, A.15 A.16, and A.8 (plus the same operations for the derivation of the v component of $\widehat{\psi}_R$). In practice, this is achieved in twice the time necessary to reconstruct the map *online*.

References

[1] S. Schaefer, T. McPhail, J. Warren, Image deformation using moving least squares, ACM Trans. Graph. (SIGGRAPH) 25 (2006) 533–540.