Rigid Motion Mesh Morpher (R3M): a novel approach for mesh deformation

George Eleftheriou[1] , Guillaume Pierrot[2]

[1]ESI Group & PhD candidate @ NTUA
[2]ESI Group

george.eleftheriou@esi-group.com

June 4, 2014

Outline

1. Morphing as a step towards automation
2. Existing mesh morphing methods
3. Brief introduction to R3M
4. Results
5. Conclusions / Next Steps / improvements

Morphing as a step towards automation

Re-meshing

Trashing of the mesh at the end of each optimization cycle and generation of a new one. Time-consuming, gradient consistency lost from one cycle to the other. In some cases manual intervention during mesh generation.

Morphing as a step towards automation

Re-meshing

Trashing of the mesh at the end of each optimization cycle and generation of a new one. Time-consuming, gradient consistency lost from one cycle to the other. In some cases manual intervention during mesh generation.

Morphing

Deformation of the existing mesh. Aim: adjoint-based optimization at iso-connectivity. Challenges: avoiding twisted/heavily distorted cells, robustness issues (mesh anisotropy, mesh rotation).

Morphing as a step towards automation

Re-meshing

Trashing of the mesh at the end of each optimization cycle and generation of a new one. Time-consuming, gradient consistency lost from one cycle to the other. In some cases manual intervention during mesh generation.

Morphing

Deformation of the existing mesh. Aim: adjoint-based optimization at iso-connectivity. Challenges: avoiding twisted/heavily distorted cells, robustness issues (mesh anisotropy, mesh rotation).

The basic idea

The internal nodes of the mesh should gracefully follow the movement of boundary nodes, as indicated by the optimization algorithm.

Existing mesh morphing methods

Method	Shortcomings
Spring analogy	Not robust
Laplacian smoothening	More robust. No rotation. No mesh anisotropy.
Linear elasticity	More robust but mesh anisotropy?
Radial Basis Functions	Dense matrices, Limitations in mesh size, trade-off between computational cost & implementation simplicity

Common characteristic of the first three: they don't handle naturally mesh anisotropy.

Brief introduction to R3M

Why "Rigid Motion" ?

Technically speaking it's not "rigid". It's "as-rigid-as-possible". And it's not meant for the entire mesh (how could it be?). It's meant for groups of nodes called **stencils**.

Brief introduction to R3M

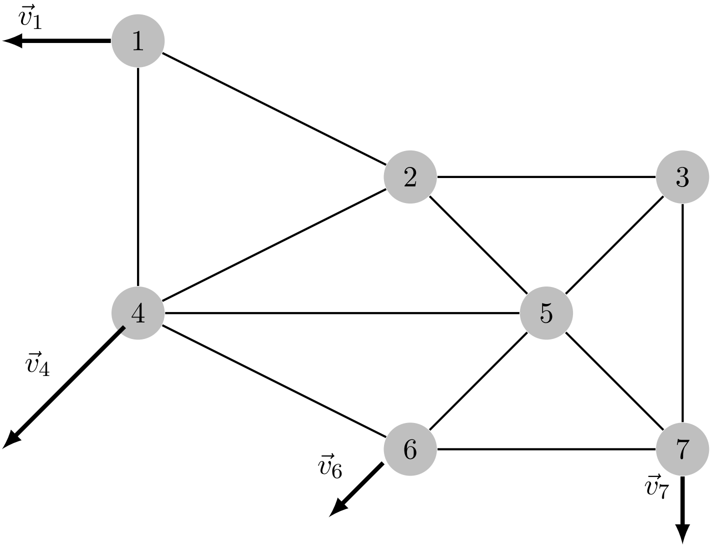
Why "Rigid Motion" ?

Technically speaking it's not "rigid". It's "as-rigid-as-possible". And it's not meant for the entire mesh (how could it be?). It's meant for groups of nodes called **stencils**.

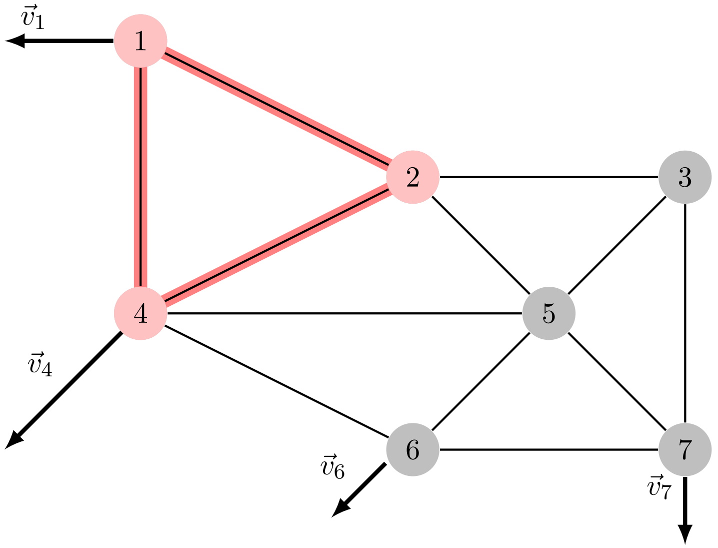
Stencils?

Example: A node plus its neighbouring nodes (sharing one or more cells with it).

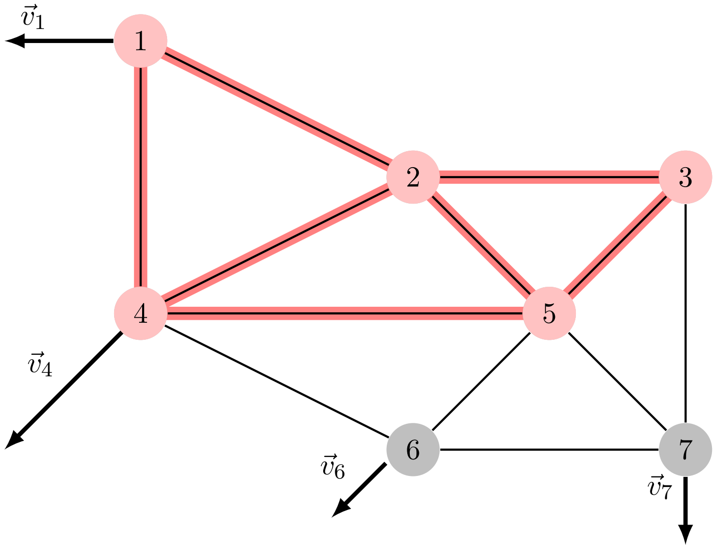
Brief introduction to R3M



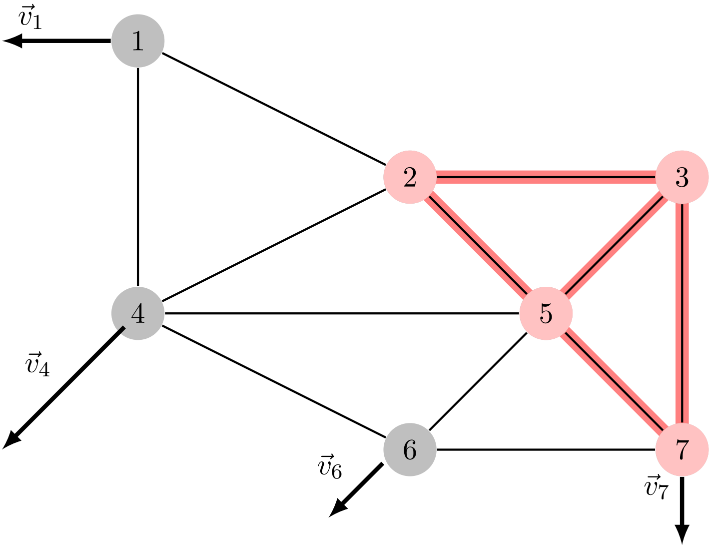
Brief introduction to R3M



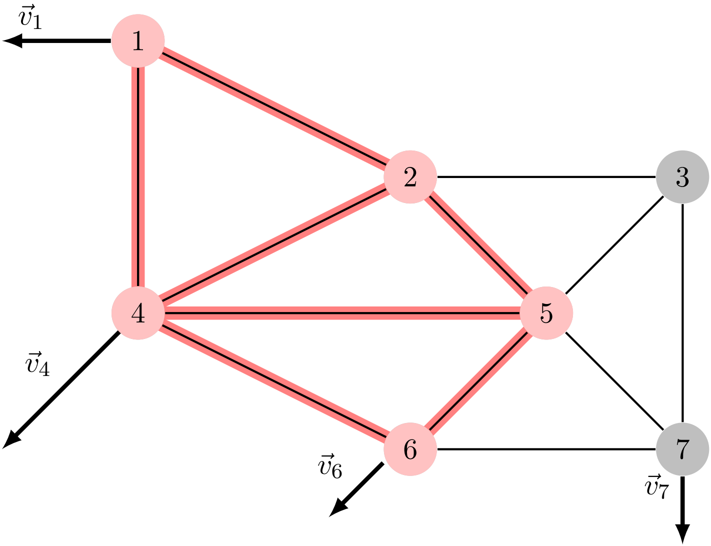
Brief introduction to R3M



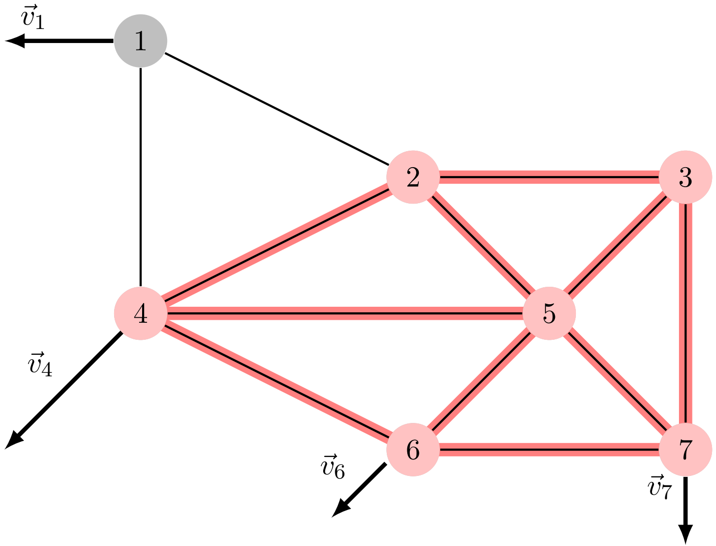
Brief introduction to R3M



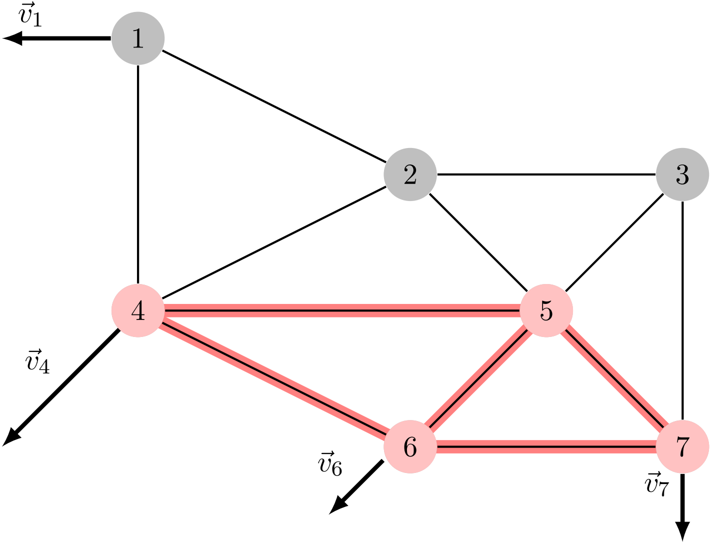
Brief introduction to R3M



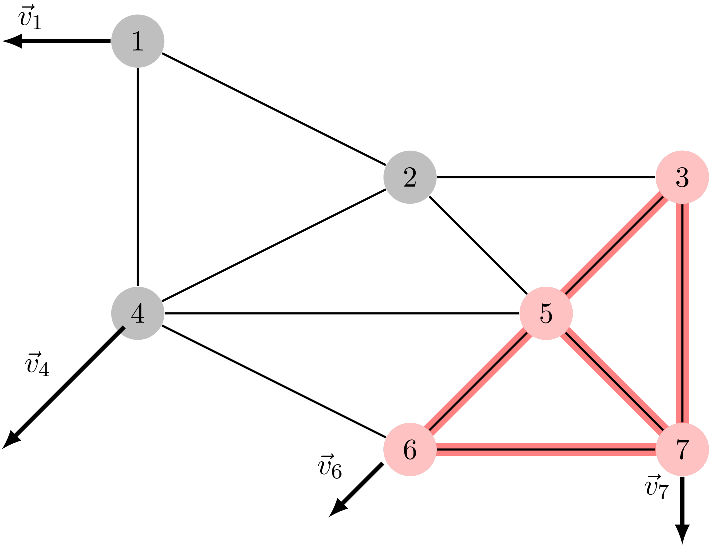
Brief introduction to R3M



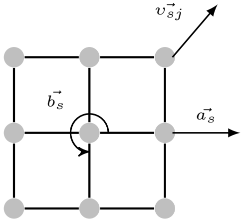
Brief introduction to R3M



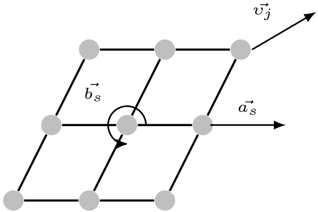
Brief introduction to R3M



Brief introduction to R3M

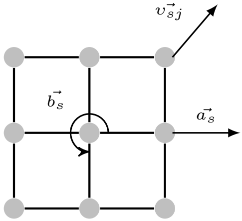


(a)
$\vec{v_{sj}} = \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$

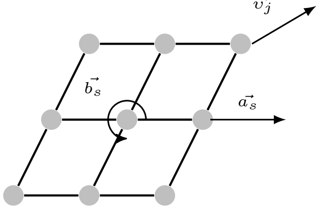


(b) $\vec{v_j} \neq \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$

Brief introduction to R3M



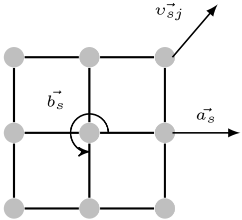
(c)
$\vec{v_{sj}} = \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$



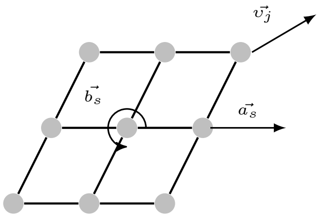
(d) $\vec{v_j} \neq \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$

$$\vec{e_{sj}} = \vec{v_j} - \vec{v_{sj}} = \vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

Brief introduction to R3M



(e)
$\vec{v_{sj}} = \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$



(f) $\vec{v_j} \neq \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$

$$\vec{e_{sj}} = \vec{v_j} - \vec{v_{sj}} = \vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

$$\vec{e_{sj}} = \sqrt{w_s \mu_{sj}} (\vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s}))$$

Brief introduction to R3M

The total distortion energy of the mesh will be

$$E = \sum_{s} \sum_{j \in s} \vec{e_{sj}}^T \cdot \vec{e_{sj}} = \sum_{s} \sum_{j \in s} \|\vec{e_{sj}}\|^2$$

It serves as the distortion metric which needs to be minimized. Hence the classification of the method as "optimization based".

$$\frac{\partial E}{\partial \upsilon_j} = \frac{\partial E}{\partial a_s} = \frac{\partial E}{\partial \beta_s} = 0$$

Final system of equations

The quadratic minimization problem of the total distortion energy of the mesh, as shown above, brings us to the following symmetric positive definite system

$$\begin{bmatrix} \mathbb{A}_{uu} & \mathbb{A}_{u(a|b)} \\ \mathbb{A}_{(a|b)u} & \mathbb{A}_{(a|b)(a|b)} \end{bmatrix} \cdot \begin{bmatrix} u \\ (a|b) \end{bmatrix} = \begin{bmatrix} P_u \\ P_{(a|b)} \end{bmatrix}$$

where the RHS consists of the boundary conditions, namely the prescribed nodes' velocities. Attempting to solve it using the Schur complement leads to two different cases of elimination:

$$\text{Either} \quad u = f(a|b) \quad \text{or}$$
$$(a|b) = g(u)$$

Final system of equations

for instance

$$u = -\mathbb{A}_{uu}^{-1} \cdot \mathbb{A}_{u(a|b)} \cdot (a|b) + \mathbb{A}_{uu}^{-1} \cdot P_u \rightarrow$$
$$(-\mathbb{A}_{(a|b)u} \cdot \mathbb{A}_{uu}^{-1} \cdot \mathbb{A}_{u(a|b)} + \mathbb{A}_{(a|b)(a|b)}) \cdot (a|b) = -\mathbb{A}_{(a|b)u} \cdot \mathbb{A}_{uu}^{-1} \cdot P_u + P_{(a|b)}$$

or

$$(a|b) = -\mathbb{A}_{(a|b)(a|b)}^{-1} \cdot \mathbb{A}_{(a|b)u} \cdot u + \mathbb{A}_{(a|b)(a|b)}^{-1} \cdot P_{(a|b)} \rightarrow$$
$$(\mathbb{A}_{uu} - \mathbb{A}_{u(a|b)} \cdot \mathbb{A}_{(a|b)(a|b)}^{-1} \cdot \mathbb{A}_{(a|b)u}) \cdot u = P_u - \mathbb{A}_{u(a|b)} \cdot \mathbb{A}_{(a|b)(a|b)}^{-1} \cdot P_{(a|b)}$$

Explanation of the μ_{sj} coefficient

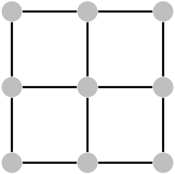


Figure: Isotropic stencil $\mathbb{T} = \mathbb{I}$

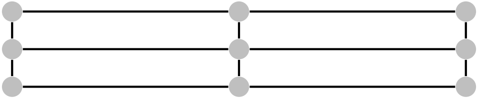


Figure: Squeezed/anisotropic stencil (we favour rigidity in the direction of squeeze) $\mathbb{T} \neq \mathbb{I}$

Explanation of the μ_{sj} coefficient

Reminder:

$$\vec{e_{sj}} = \sqrt{w_s \mu_{sj}}(\vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s}))$$

Relative weight stencil $\Rightarrow$ node (scalar coefficient):

$$\mu_{sj} = \frac{\exp\left(-\frac{\|\vec{x_j} - \vec{c_s}\|^2}{h^2}\right)}{\|\vec{x_j} - \vec{c_s}\|^2 + \varepsilon}$$

Results

Conclusions / Next Steps / improvements

R3M is/does:

- Essentially mesh-less (only needs nodes and not cell or inertial data)
- Manage intrisically mesh anisotropy and rotation.

Better weighting

Improved propagation of deformation to all layers by setting the coefficients $w_s = f(\int_t E_s dt)$ as a function of the integral of the stencil distortion energy over time.

Coupling with ESI's i-adjoint solver

for automated optimization loops

Thank you

This work was funded by the EU through the FP7-PEOPLE-2012-ITN "AboutFlow" Grant agreement number:317006.