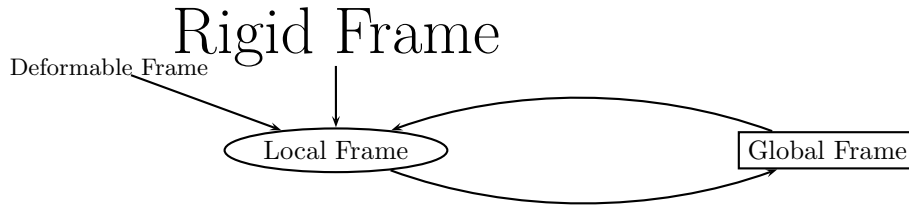


Deformable On Rigid Mapping

Motivation: Rigid bodies kinematics and dynamics are already modeled in Sofa, but in reality mechanical objects could be a non perfect rigid. It suggest to create a rigid model allowing a bit of deformation. The idea for computation is the same one of Rigid Mapping combining with a deformable local frame.



Definitions:

- Model Behaviour Rigid : Position and orientation (quaternion) of the center(s) of mass.
- Model Colision Rigid : A set of particles used for computing the colision with other objects.
- Model Deformable Rigid : A set of particle used for modeling the deformation.

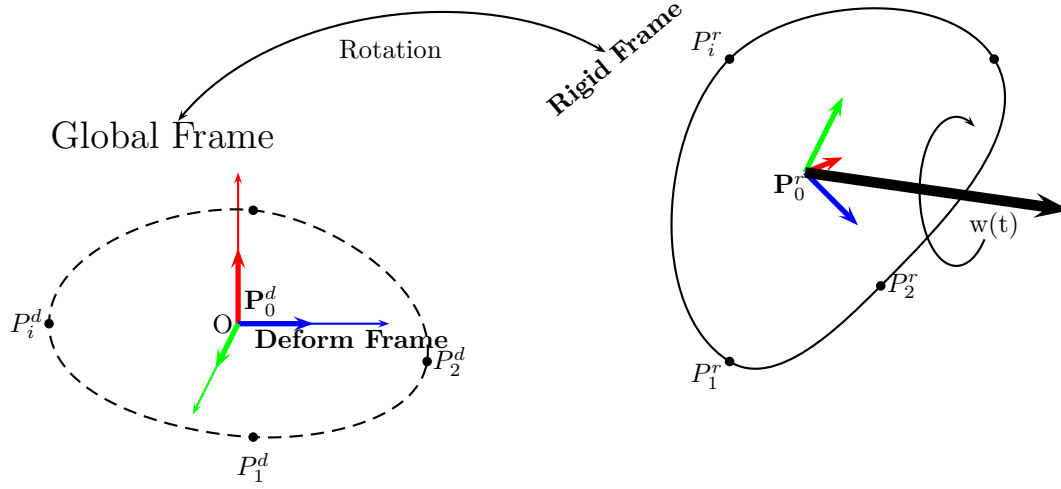
It is needed to well understand the Rigid Mapping for good understanding the Deformable On Rigid Mapping.

Computation: Generalized from the perfect Rigid model, the Deformable Rigid model is more complicated, but consist also updating the particles position, velocity and computing the effect of the external forces. The Deformable Rigid has tree frame of reference : a global frame and a local frame like in the perfect Rigid, a further local frame. The unique aim of the added frame is modeling the deformation, so this frame will be fixed in relation to the global frame and don't have any rotation or translation. To find the particles positions, the center position must be found at first. The other ones will be found locally from the deformed system¹² : $OP_i^r = OP_0^r + P_0^d P_i^d$. To find the particles velocities, not only the velocity due to the rigid center motion (v^r) must be computed, but also the velocity due to the deformation (v^d) must be computed and added. For each particle : $v_i = v_i^d + v_i^r$. External forces applied both to the rigid center and to the deformable model :

$$\sum F_{extern} + \rightarrow \begin{cases} \text{center.} \\ \text{deformable system.} \end{cases}$$

¹[...] ^d denote all on the deformable frame

²[...] ^r denote all on the rigid frame



To update the particles position :

```

|| Finding  $OP_o^r = LinearPosition$ ; Finding  $R_{loc}^{glob} = RotationMatrix$ ;
|| For( $i = 1; i < N; i++$ )
|| {
||      $P_o^r P_i^r = R_{loc}^{glob} * P_o^d P_i^d$ ;
||      $OP_i^r = OP_o^r + P_o^r P_i^r$ ;
|| }

```

To update the particles velocity :

```

|| Finding  $V_{Po}^r = LinearVelocity$ ; Finding  $w_{Po}^r = AngularVelocity$ ;
|| For( $i = 1; i < N; i++$ )
|| {
||      $V_i^r = V_{Po}^r - P_o^r P_i^r \wedge w_{Po}^r$ ;
||      $R = RotationMatrix$  ;  $V_i^d = R * V_{P_i}^r$ ;
||      $V_i = V_i^r + V_i^d$ ;
|| }

```

To compute the reponse of the external forces:

```

|| For( $i = 1; i < N; i++$ )
|| {
||      $F_{exter} += f_i$ ;
||      $Torque += P_o P_i \wedge f_i$ 
||      $R = [RotationMatrix]^{-1}$  ;  $F_{P_i^d} = R * f_i$ ;
|| }
|| Update  $LinearForceCenter = F_{exter}$ ;
|| Update  $AngularForceCenter = Torque$ ;

```

Implemented code in Sofa: These algorithms are implemented in Sofa by the following methods :

To update the particles position :

```
template <class BasicMapping>
void DeformableOnRigidFrameMapping<BasicMapping>::apply(typename Out::VecCoord & out,
                                                         const typename In::VecCoord & inDeformed,
                                                         const typename InRoot::VecCoord * inRigid)
```

To update the particles velocity :

```
template <class BasicMapping>
void DeformableOnRigidFrameMapping<BasicMapping>::applyJ(typename Out::VecDeriv & out,
                                                         const typename In::VecDeriv & inDeformed,
                                                         const typename InRoot::VecDeriv* inRigid)
```

To compute the external forces soliciting to center :

```
template <class BasicMapping>
void DeformableOnRigidFrameMapping<BasicMapping>::applyJT(typename In::VecConst & out,
                                                         const typename Out::VecConst & in,
                                                         typename InRoot::VecConst* outroot)
```

Sofa Keyword: Rigid Mapping, Deformable on Rigid Frame Mapping.

Document written by
Chi Thanh NGUYEN
INRIA Lille