

*Simon COURTEMANCHE
Xavier MARCHAL*

*Antoine KUHN
Bertrand RICHARD*

Mise en correspondance de deux maillages bruités

Projet de spécialité Image

Le 19 juin 2009

Encadrants : Franck Hétroy, Romain Arcila



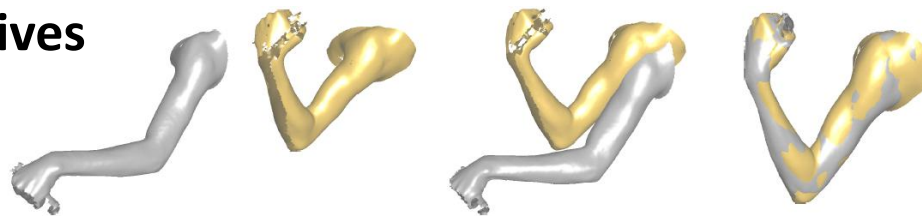
Sommaire



Introduction

1. Installation des logiciels – Prise en main du scanner (1^{ière} semaine)
2. Implémentation de l'algorithme : Non-rigid Registration
 - a. Organisation de l'équipe
 - b. Résumé des différentes étapes :
 - i. Difficultés rencontrées
 - ii. Résultats
3. Assemblage final – Interface

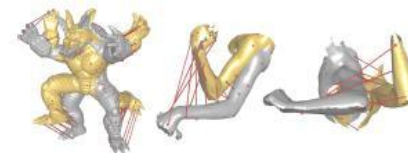
Conclusion - Perspectives



Installation des logiciels – Prise en main du



scanner (1^{ière} semaine)



Logiciels :

– Open mesh



– QGLViewer



– QT

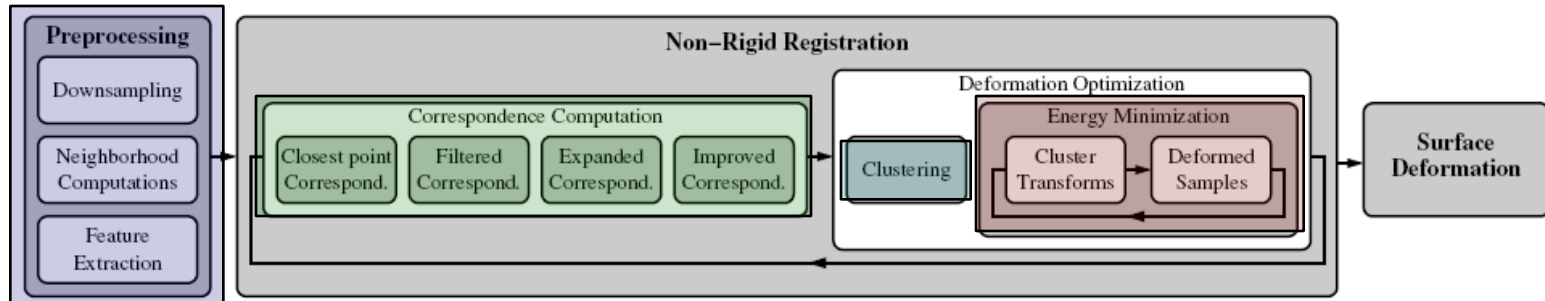


Scanner :



Implémentation de l'algorithme

Q. Huang et al. / Non-Rigid Registration



- a. Organisation de l'équipe
- b. Résumé des différentes étapes :
 - i. Difficultés rencontrées
 - ii. Résultats

Implémentation : Preprocessing

- 3 étapes :
 - 1 : Downsampling (étape abandonnée)

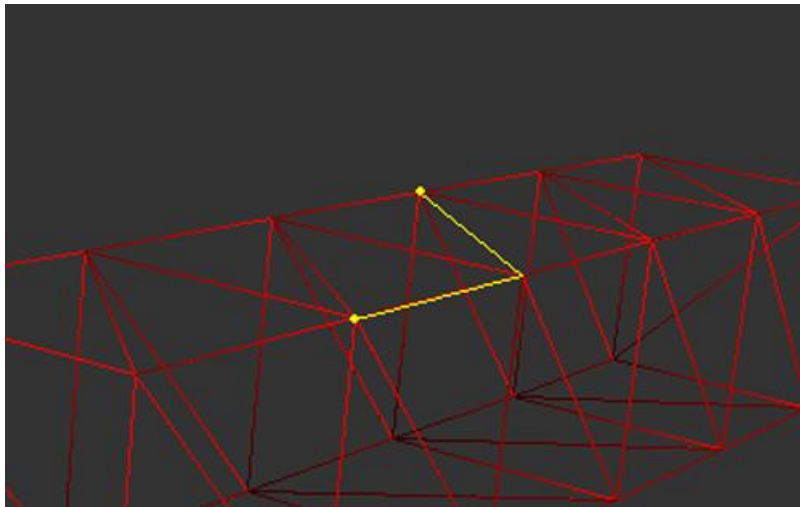


7M polygons

Octree Simplification (L2 err = 0.001) VS-Tree Simplification (same error)

Implémentation : Preprocessing

- 2 : Geodesic distance & Neighborhood
Distance en passant par les arêtes

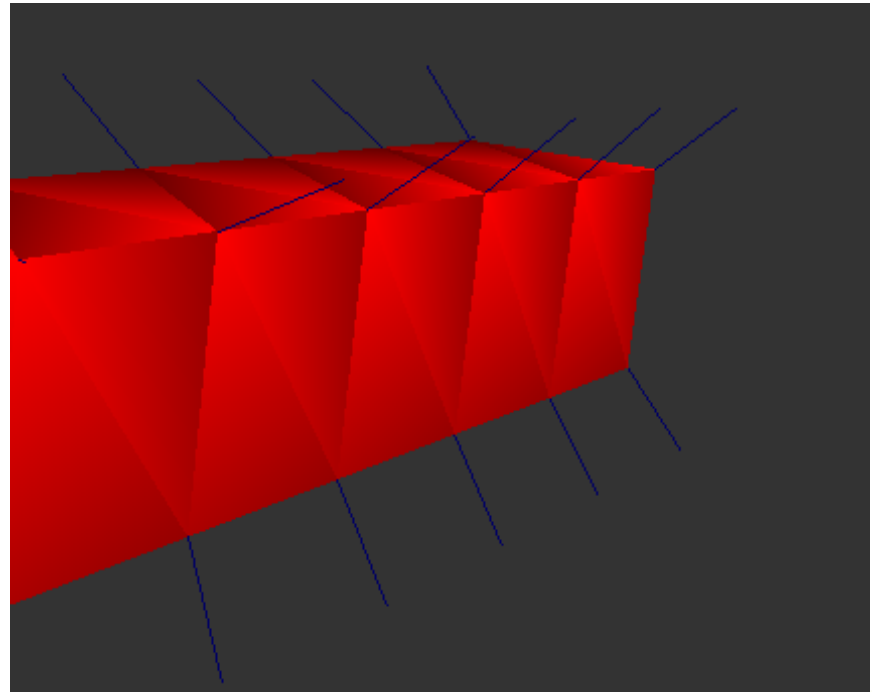


Voisinage : k plus proches voisins, $k = 15$

Implémentation : Preprocessing

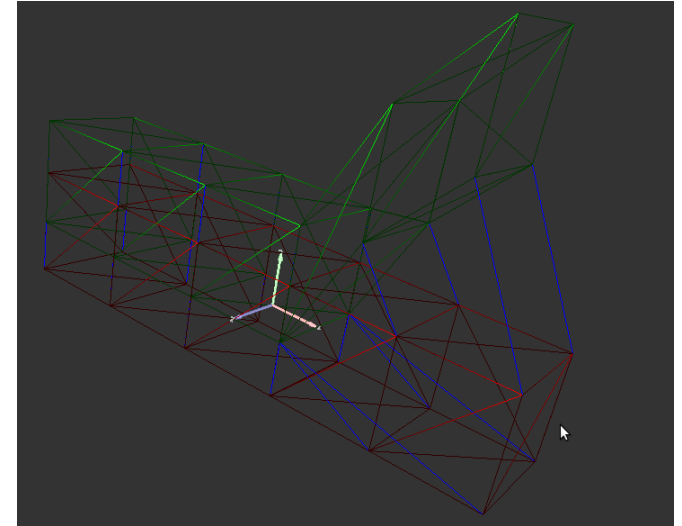
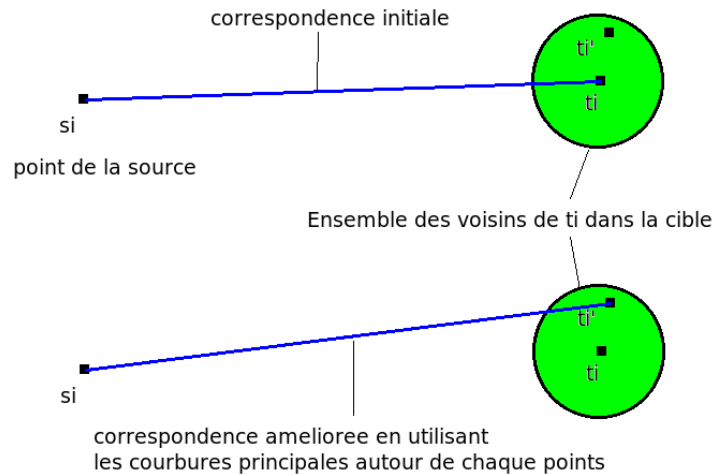
- 3 : Normals & Feature vector

On estime les courbures principales à partir des normales au point et à celles de son voisinage



Implémentation : Correspondence

- Initial correspondence :
 - Closest Points
 - Improved correspondences

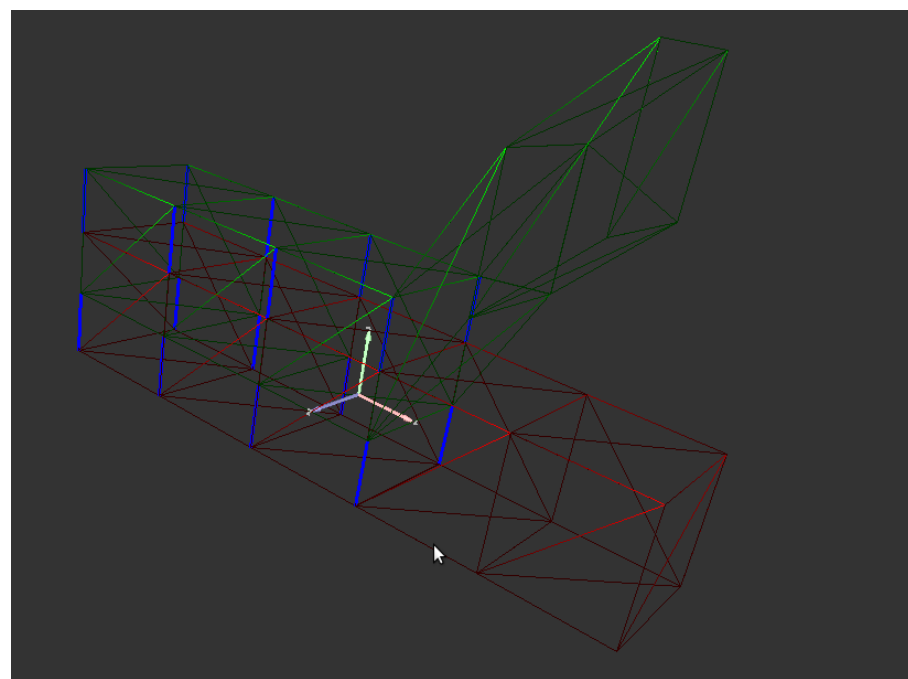


Implémentation : Correspondence

- Pruning correspondences :

$$c_{ij} = \min \left\{ \frac{d_g(\mathbf{s}_i, \mathbf{s}_j)}{d_g(\mathbf{t}_i, \mathbf{t}_j)}, \frac{d_g(\mathbf{t}_i, \mathbf{t}_j)}{d_g(\mathbf{s}_i, \mathbf{s}_j)} \right\},$$

$$M_{ij} = \begin{cases} \left(\frac{c_{ij} - c_0}{1 - c_0} \right)^2 & c_{ij} > c_0, \\ 0 & \text{otherwise,} \end{cases}$$

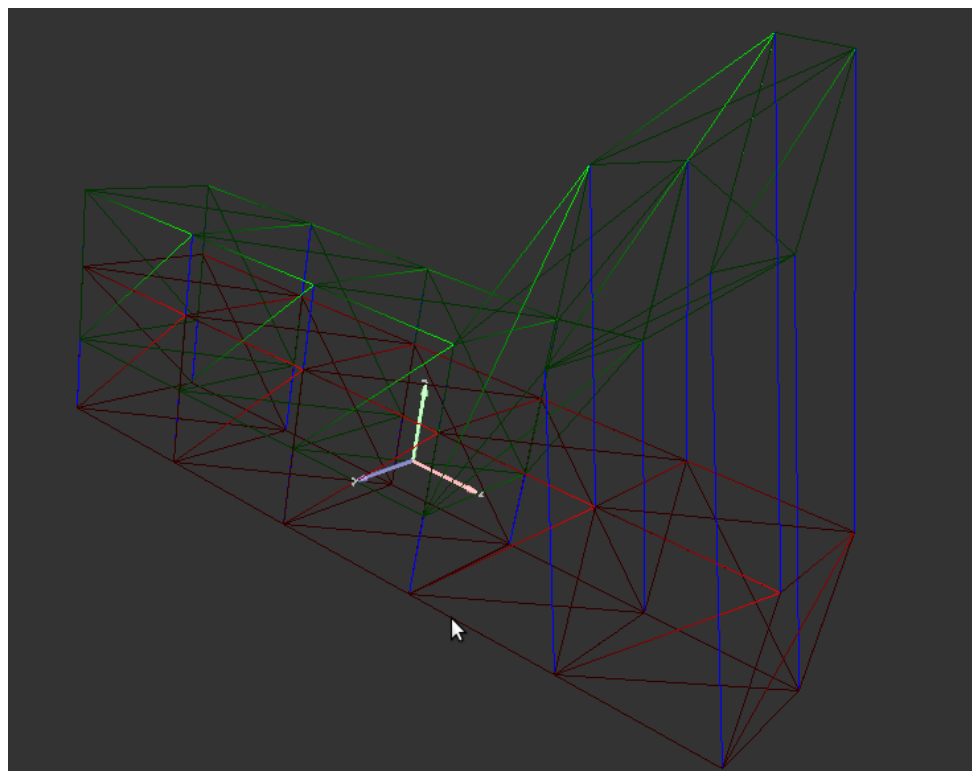


Implémentation : Correspondence

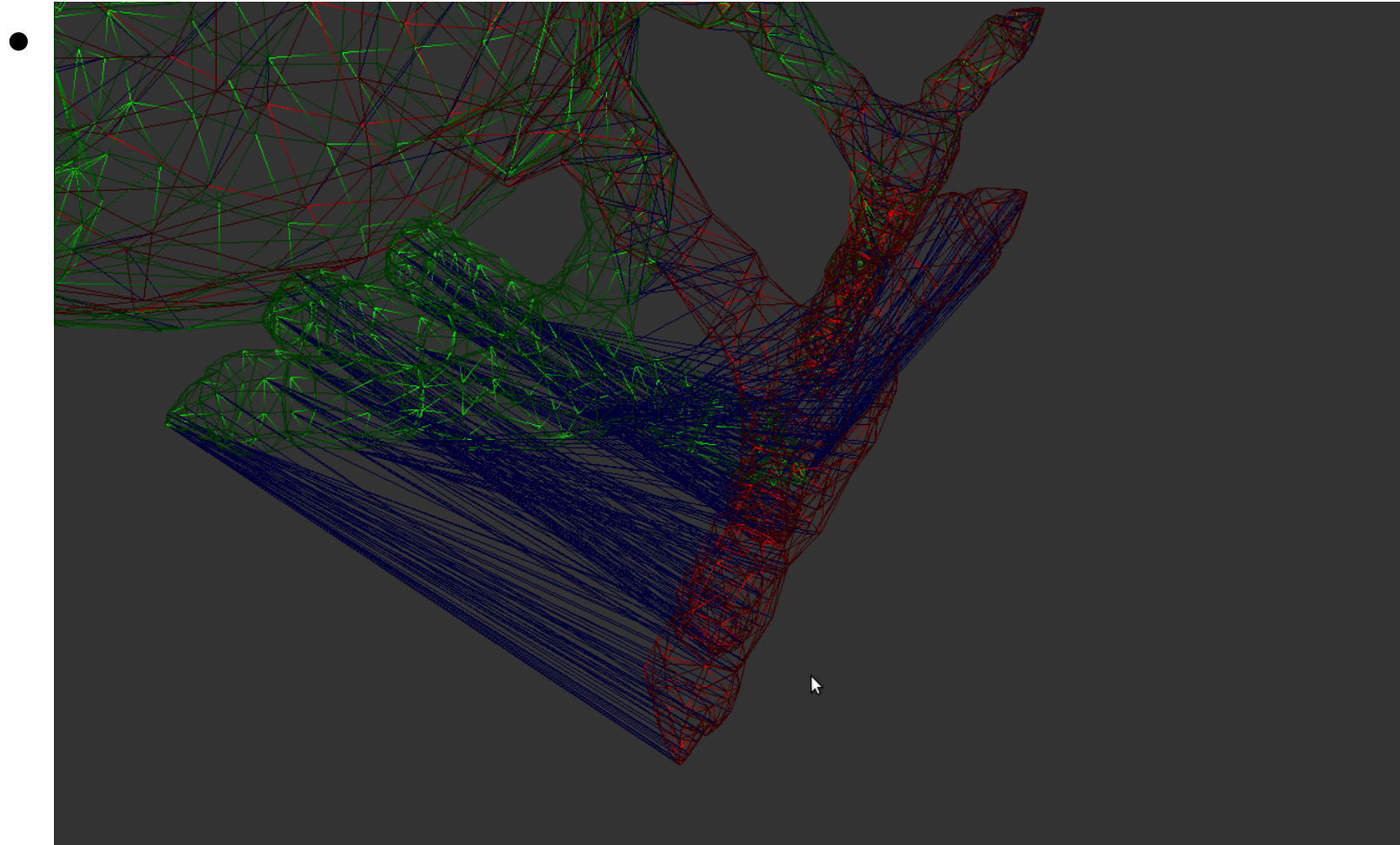
- Propagation correspondences :

$$\mathbf{t}_j = \operatorname{argmin}_{\mathbf{t} \in \mathcal{N}_g(\mathbf{t}_j, \mathcal{T})} e_{\mathcal{K}}(\mathbf{s}_i, \mathbf{t})$$

$$e_{\mathcal{K}}(\mathbf{s}, \mathbf{t}) = \sum_{(\mathbf{s}_k, \mathbf{t}_k) \in \mathcal{K}} [d_g(\mathbf{s}, \mathbf{s}_k) - d_g(\mathbf{t}, \mathbf{t}_k)]^2,$$

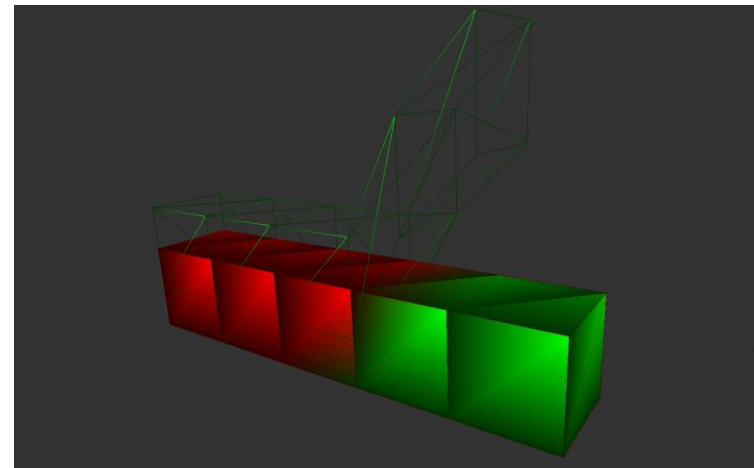
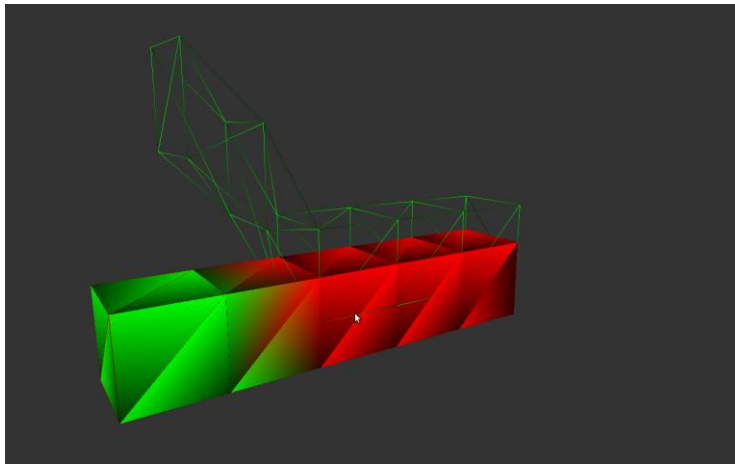


Implémentation : Correspondence



Implémentation : Clustering

- *Principe* : recherche des parties du maillage qui peuvent avoir une transformation « rigide » identique.

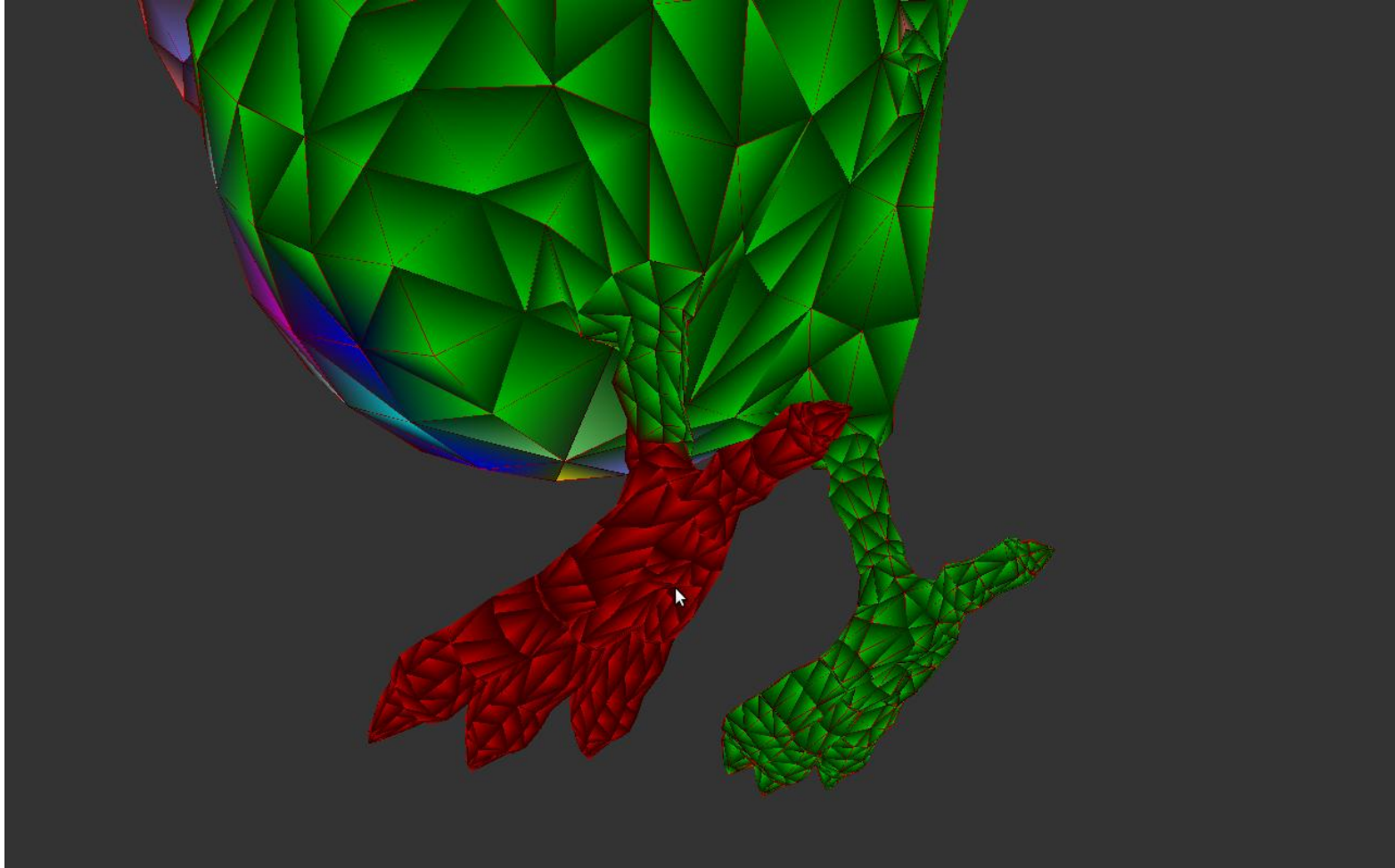


- *Initialisation* : Chaque point représente un cluster
- *Traitement* : on essaie de fusionner le maximum de clusters entre eux.

Implémentation : Clustering

- *Algorithme* :
 - Chaque cluster possède une transformation rigide associée (calculée sur son *cluster étendu*) et une erreur de transformation.
 - Boucle itérative : Pour chaque cluster C, on regarde les clusters voisins et on regarde l'erreur associée si on applique la transformation de C. Si elle est inférieure à un seuil, on fusionne.
- *Problème* :
 - Quelle valeur donne-t-on au seuil?
 - Notions de voisins?

Implémentation : Clustering

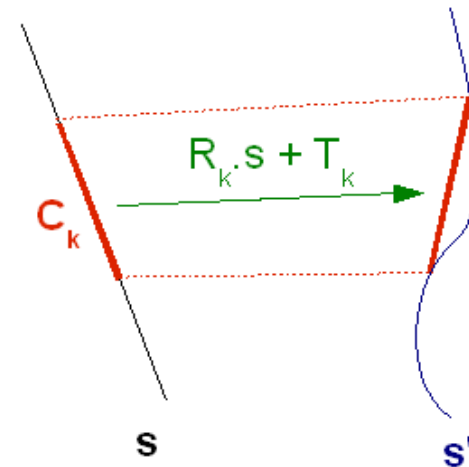
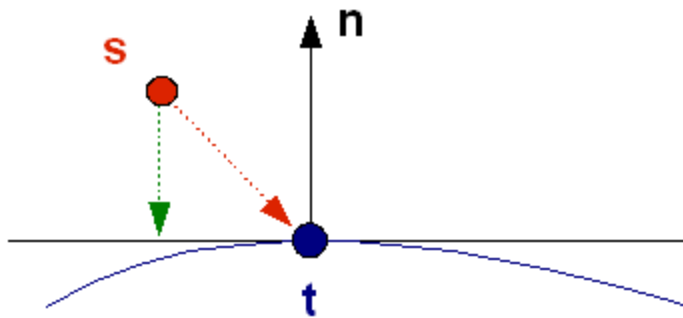


Implémentation : Energy minimization

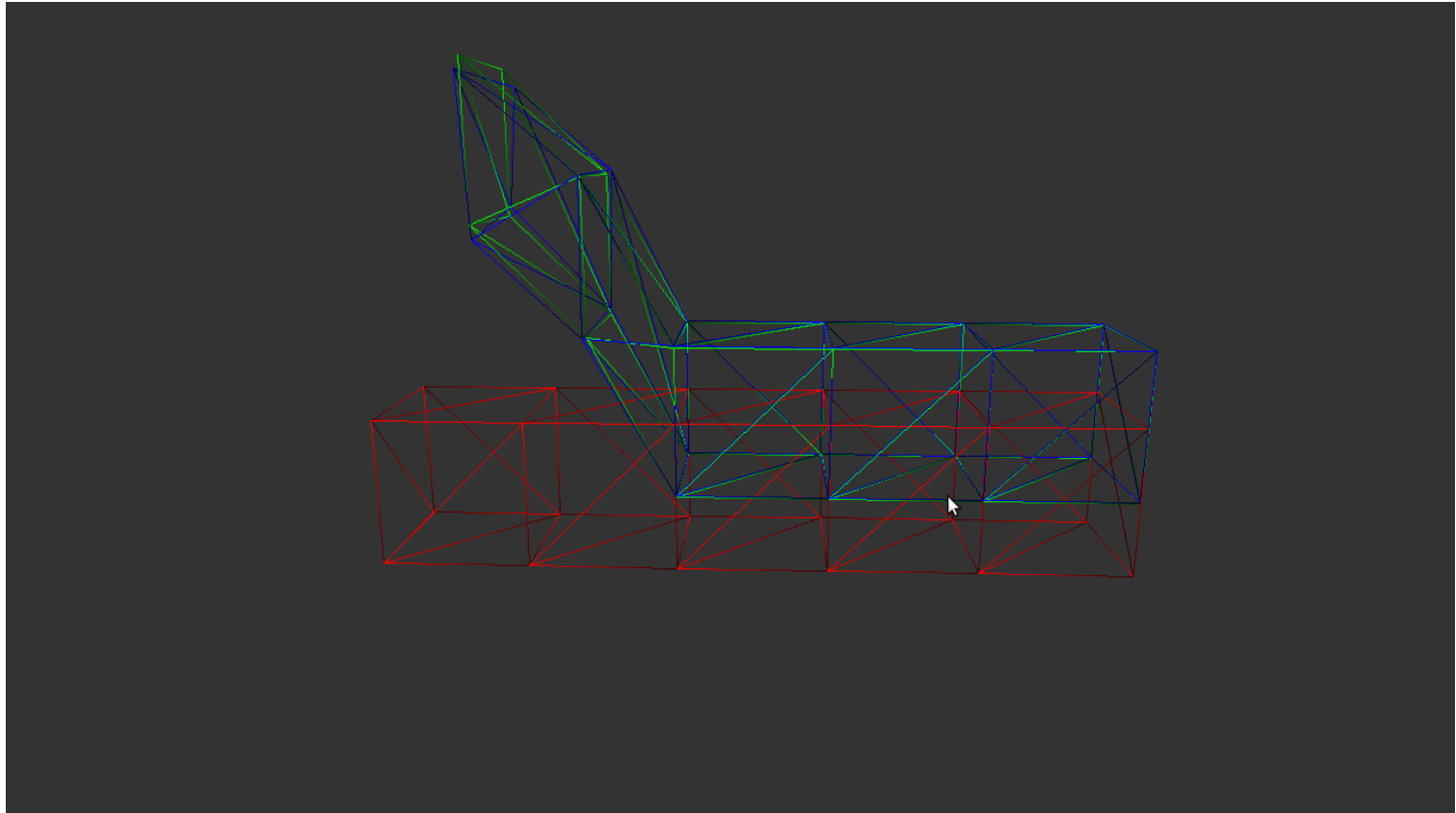
$$E = \lambda_{\text{corr}} E_{\text{corr}} + \lambda_{\text{rigid}} E_{\text{rigid}}$$

$$E_{\text{corr}} = \sum_{s_i \in \hat{S}} w_i \left[\alpha \|s'_i - t_i\|^2 + \beta ((s'_i - t_i)^T \mathbf{n}_i)^2 \right]$$

$$E_{\text{rigid}} = \sum_k E_k = \sum_k \sum_{s_i \in \tilde{C}_k} \|\mathbf{R}_k s_i + \mathbf{T}_k - s'_i\|^2$$



Implémentation : Energy minimization



Assemblage final – Interface

- Création d'une interface de visualisation
 - Visualise des maillages
 - Visualise les clusters, normales
 - De nombreuses options (cf. démonstration)

Conclusion - Perspectives

- Bilan : Ce qu'il reste à faire + Impressions
- Perspectives du projet
 - Optimisation du code C++
 - Gestion des étapes « abandonnées » (pour tester sur des maillages plus importants)

Merci de votre attention !