

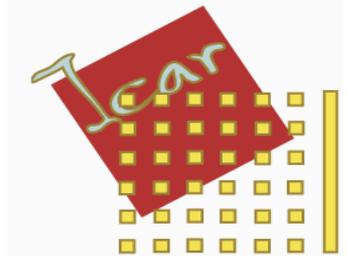
Cécile Peladan

M2 Informatique spécialité IMAGINA

# Développement de nouvelles fonctions dans le logiciel MeshTools : optimisation de l'affichage 3D, recalage automatique de maillages



**UNIVERSITÉ  
DE MONTPELLIER**



Encadrants:

Renaud LEBRUN

Gérard SUBSOL

03 Septembre 2015

# Sommaire:

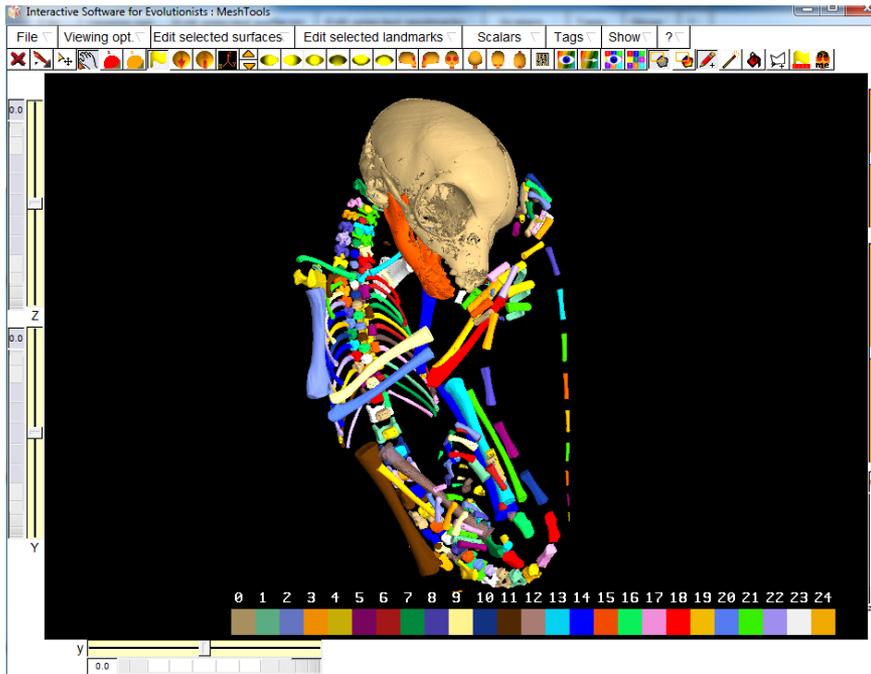
- I. Sujet
- II. Optimisation de l'affichage 3D
  - 1. Affichage
  - 2. Résultats du Vertex Buffer Object
- III. Recalage
  - 1. Algorithme ICP (Iterative closest point) (Rigide, Similitude, Affine)
  - 2. Algorithme ICP non rigide (Localement Affine)
  - 3. Résultats du recalage
- IV. Conclusion et perspectives

# I. Sujet

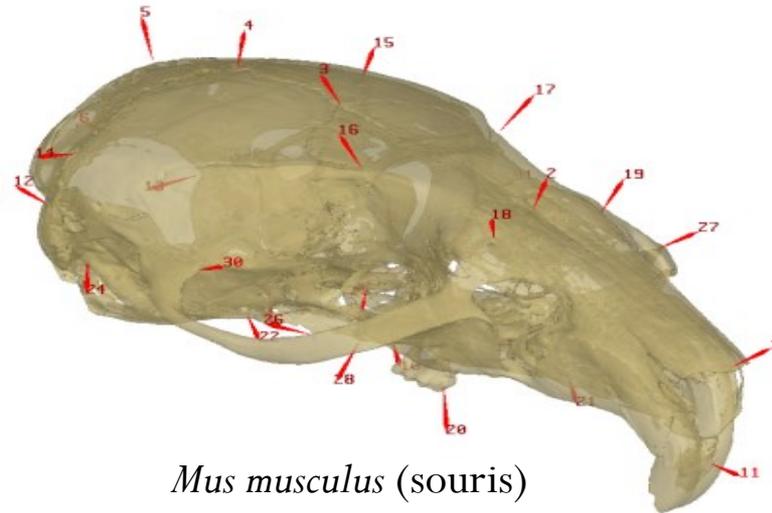
# Contexte du sujet:

- Le logiciel ISE-MeshTools.

<http://morphomuseum.com/meshtools>



*Lemur catta* (maki)

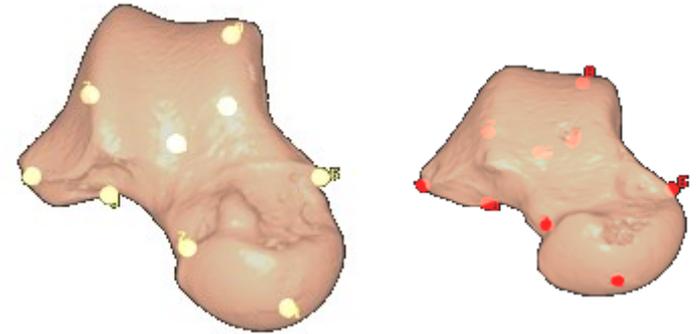


*Mus musculus* (souris)

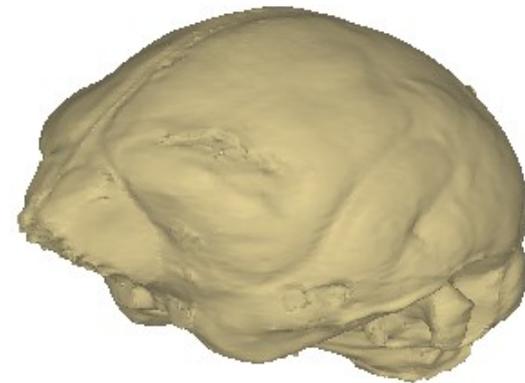
Maillages 3D

# Comparaisons des maillages

- Comparaison de structures anatomiques
- Difficulté d'établir des appariements entre deux maillages (cavité endocrânienne, astragale (*Talus*), ...) en particulier des zones planes, sans foramen...



Astragales



Cavité endocrânienne

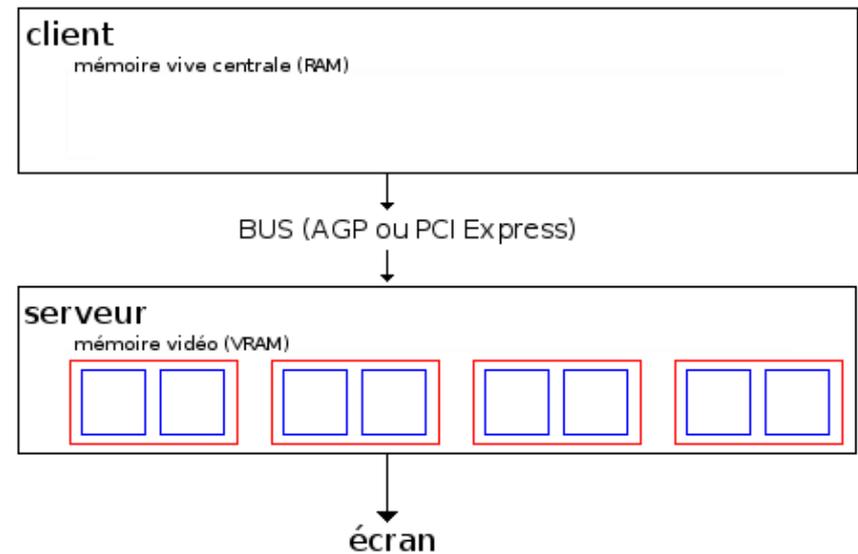
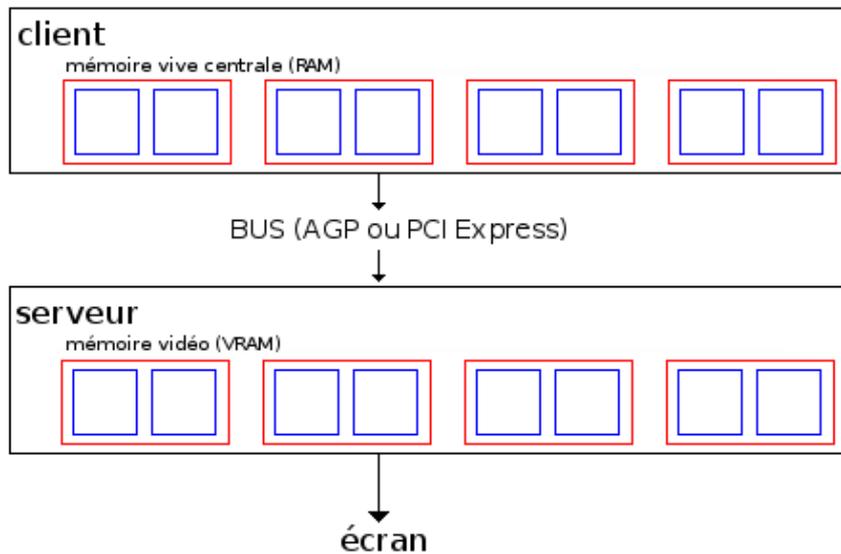
# Cahier des charges:

- Ajouter deux fonctionnalités au logiciel :
  - Améliorer l'affichage 3D, en utilisant les fonctionnalités GPU les plus avancées d'OpenGL, (25 fps).
  - Implémenter une fonction de mise en correspondance automatique sur toute la surface.

## II. Optimisation de l'affichage 3D

# Affichage

- MeshTools a des fonctions immédiates (*GLBegin*, *GLEnd*, *GLVertex3f*, ...).
- Vertex Array et Vertex Buffer Object :



Gain de performance de Vertex Buffer Object

# Implémentation

## Fonction immédiate

À chaque itération:

- glBegin(GL\_TRIANGLES);
  - Pour  $i = 0$  à nbTriangle
    - Pour  $j=0$  à 3 (points)
      - glVertex3f(x, y, z);
      - glNormal3f( $n_x$ ,  $n_y$ ,  $n_z$ );
      - glColor4f( $c_1$ ,  $c_2$ ,  $c_3$ ,  $c_4$ );
- glEnd();

## VBO

Initialisation à la 1<sup>er</sup> itération:

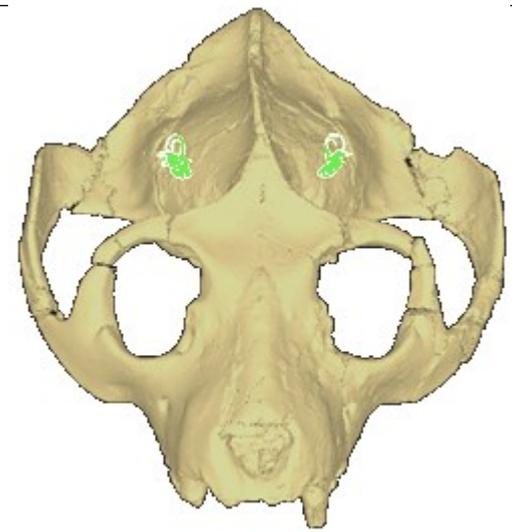
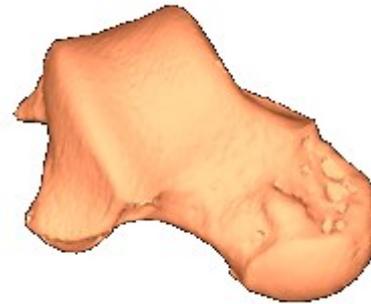
- Construction de 2 buffers:
  - Propriété des points
  - Indices des triangles

À chaque itération,

- Si détection d'une modification
  - Modifier les tableaux
  - Les copier dans les buffers
- On dessine les éléments des buffers

# Résultats du VBO:

Testé sur une machine ayant  
core i7, de 2.00GHz et  
8 Go de mémoire vive avec  
une carte graphique NVIDIA Geforce 840M.

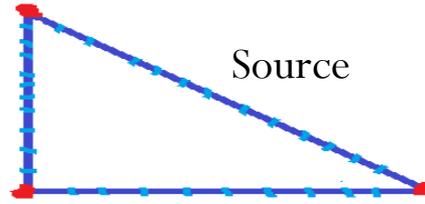
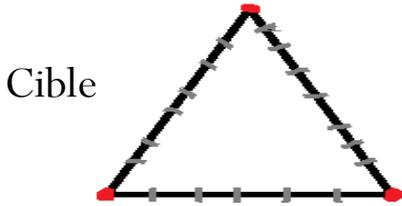


- Résultat de l'affichage en image par seconde (Frame Per Second).

Maillages	Fonction « immédiate » (en FPS)	Avec Vertex Buffer Object (en FPS)
Talus de <i>Lemur catta</i> contenant 258 663 points	8 - 9	66
Crâne de <i>Leptadapis</i> avec ses oreilles internes contenant 1.123.735 points au total	1 - 4	66

# III. Recalage 3D

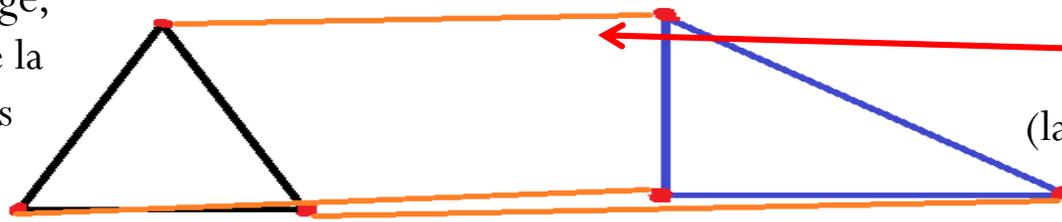
# Principe du recalage



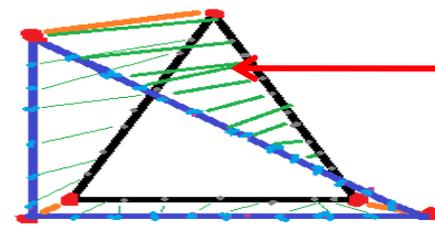
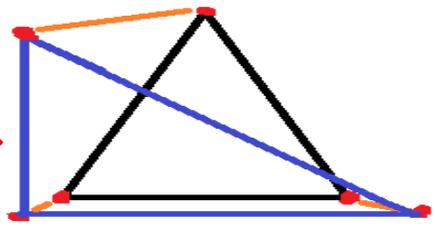
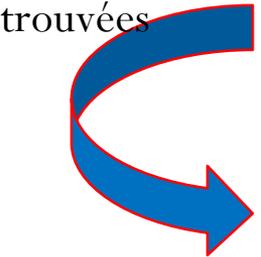
Landmarks mis par l'expert



Itération du recalage,  
Transformation de la  
source à partir des  
correspondances  
trouvées

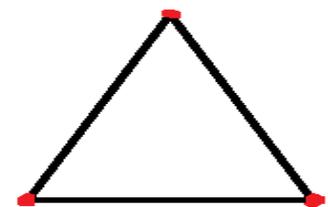
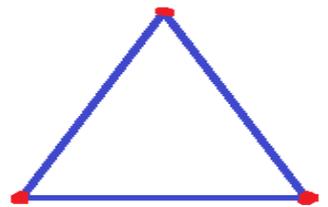
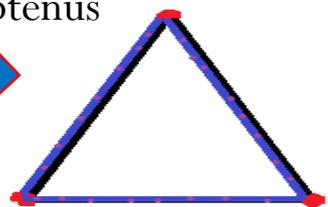


Correspondances des  
landmarks :  
(landmarks<sub>Source</sub>, landmarks<sub>Cible</sub>)



Correspondances des  
points des maillages:  
(point<sub>Source</sub>, point<sub>Cible</sub>)

Résultats  
obtenus



Source déformée

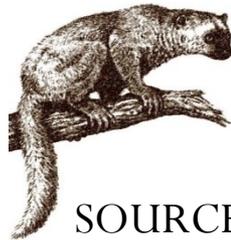
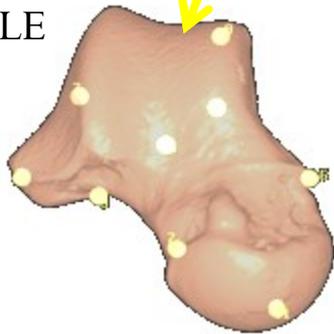
Cible

# Principe du recalage:



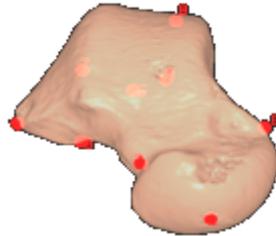
CIBLE

Landmarks  
cibles

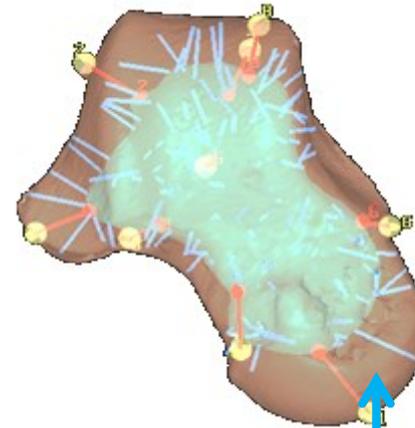
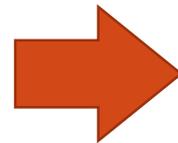


SOURCE

Landmarks  
sources

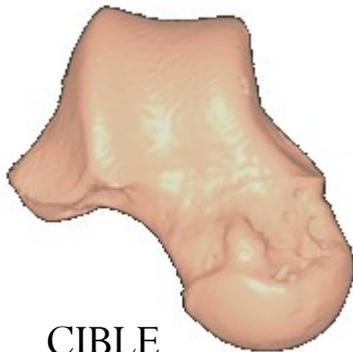


Talus d'*Adapis*

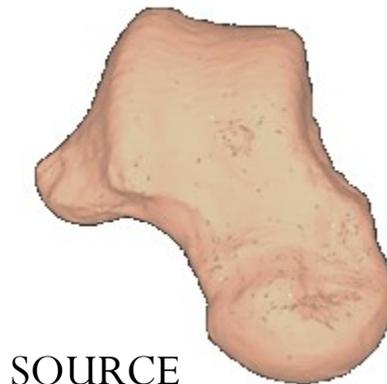


Modification  
de la source  
à chaque  
itération  
du recalage

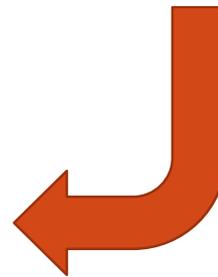
Talus de *Lemur catta*



CIBLE



SOURCE  
DEFORMEE



Les correspondances sont  
définies par :  
(point source, point cible  
le plus proche).



**points  $\neq$  landmarks**

# Algorithme ICP: Iterative Closest Point

(Paul J. Besl et Neil D. McKay, 1994)

(Zhang, 1994)

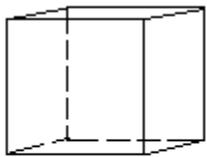
Articles:

(S. Rusinkiewicz et M. Levoy, Efficient variants of the ICP algorithm, 2001)

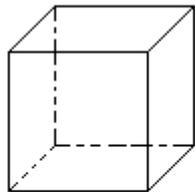
(Doug M. Boyer et al, Algorithms to automatically quantify the geometric similarity of anatomical surfaces, 2011)

# Algorithme ICP: Iterative Closest Point

1. Appariement de points en cherchant le point le plus "proche" (distance euclidienne en première approche)
2. Calcul de transformations à partir des appariements (moindres carrés par exemple)
3. Application de la transformation sur la source
4. Itérations

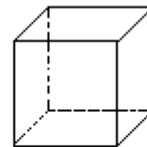


**cube initial**



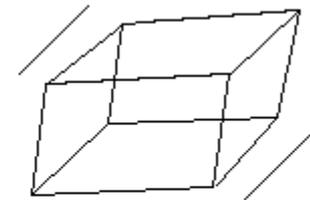
**transformation rigide**

$$[R \ t]$$



**transformation similitude**

$$[R \ t] * s$$



**transformation affine**

$$[A \ t] \text{ matrice affine}$$

# ICP : Structure

indépendant

RIGIDE

SIMILITUDE

Rigide + Mise à l'échelle

AFFINE

LOCALEMENT  
AFFINE

2 fois ICP « classique »:  
- avec les landmarks  
- avec les points sources

dépendance

1 fois ICP « classique »:  
- avec les points sources avec  
ou non ajout des landmarks

dépendance

ICP déformable:  
- seulement avec les points  
sources

Affine sans  
landmarks

Similitude

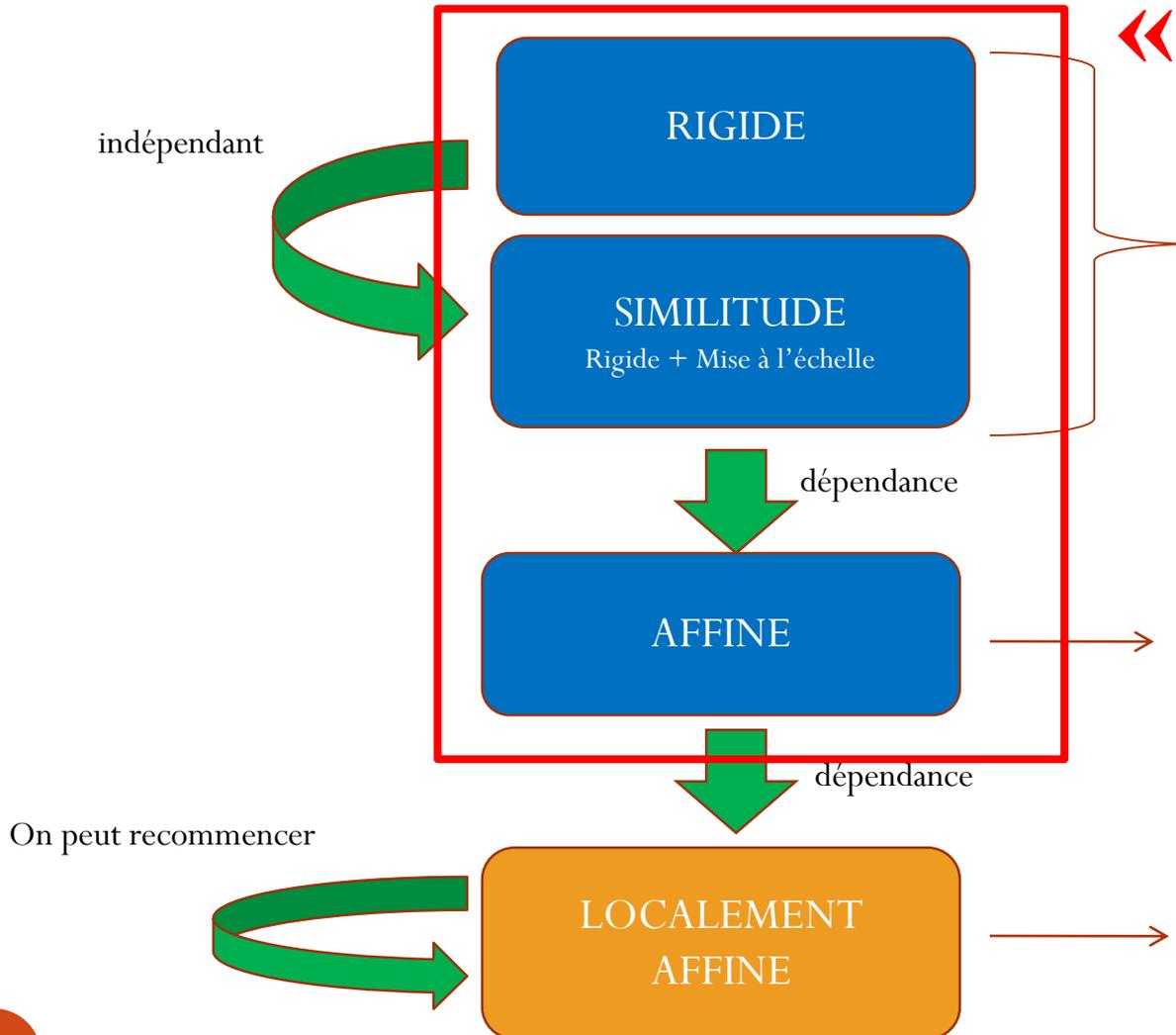
Affine avec  
landmarks

On peut recommencer

# ICP : Structure

# ICP

« classique »



2 fois ICP « classique »:

- avec les landmarks
- avec les points sources avec ou non ajout des landmarks

1 fois ICP « classique »:

- avec les points sources avec ou non ajout des landmarks

ICP déformable:

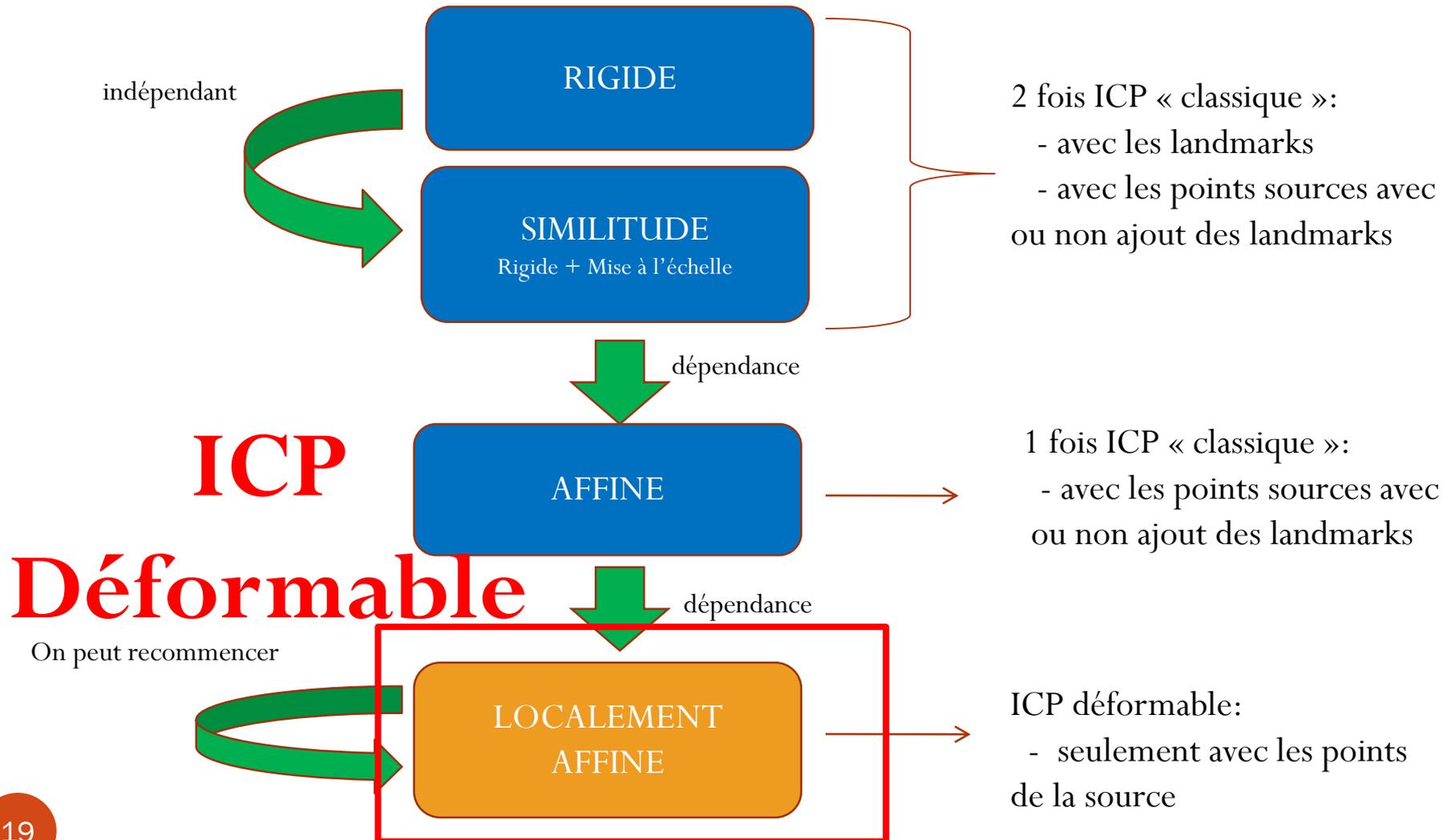
- seulement avec les points de la source

# ICP « classique »

pour mode Rigide, Similitude et Affine:

- À chaque itération,
  1. Construction des correspondances ( $\text{point}_{\text{Source}}, \text{point}_{\text{le_plus_proche_Cible}}$ ) (point à N dimensions) en utilisant du Kd-tree de la cible.
  2. Affectation d'un poids  $w_i$  à chaque correspondance trouvée  $C_i$  selon leur couleur et leur distance. Si  $w_i = 0$ , on supprime  $C_i$ .
  3. Calcul de la transformation aux moindres carrés pondérée à partir des correspondances. Application sur tout le maillage source.
  4. Calcul de la distance moyenne  $X$  d'erreur entre les nouveaux points sources et les points cibles. Si  $X >$  seuil ou le nombre maximal d'itération atteint, on arrête l'ICP.
    - les coordonnées 3D,
    - les coordonnées de leurs normales,
    - 2 courbures (minimales, maximales).

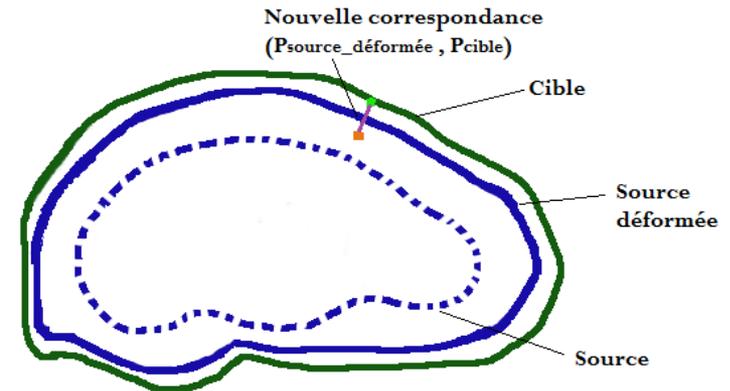
# ICP : Structure



# ICP déformable:

(J.Feldmar et N.Ayache , Rigid, Affine and Locally Affine Registration of Free-Form Surface, 1996)

- Prétraitement: Création pour chaque point  $M_k$  de la Source un sous-ensemble de points sources,  $Sphere(M_k)$  de centre  $M_k$  de rayon  $R$ .
- A chaque itération,
  1. Construction des correspondances  $(M_k, N)$  ( $N = \text{point}_{\text{le_plus_proche_Cible}}$ ).
  2. Calcul de la transformation rigide  $(R_k, t_k)$  de chaque  $M_k$  à partir des correspondances trouvées dans  $Sphere(M_k)$ .
  3. Calcul de sa transformation affine  $(A_k, b_k)$  de chaque  $M_k$ :
    - a. Récupération de  $(R_l, t_l)$  de chaque point  $M_l \in Sphere(M_k)$ .
    - b. Calcul de sa moyenne pondérée, par la distance  $\beta(M_l, M_k)$ , des transformations rigides des  $M_l \in Sphere(M_k)$ .
  4. Calcul de la distance moyenne  $X$  d'erreur entre les nouveaux points sources et les points cibles. Si  $X >$  seuil ou le nombre maximal d'itération atteint, on arrête ICP déformable.



$$\beta(M_l, M_k) = \frac{(1 - \frac{\|\vec{M}_l M_k\|}{\text{diamètre}_{\text{Source}}})}{(\sum 1 - \frac{\|\vec{M}_l M_k\|}{\text{diamètre}_{\text{Source}}})}$$

# Implémentation

- Langage en C++
- Bibliothèques VTK et FLTK
- KD TREE : Recherche du point le plus proche :
  - Distance euclidienne du Kdtree:

$$d^2(S,T) = (1 + \beta_{\text{Normale}} + \beta_{\text{Courbure}}) * (\alpha_1 * [(x-x')^2 + (y-y')^2 + (z-z')^2]) + \beta_{\text{Normale}} * (\alpha_2 * [(n_x-n_x')^2 + (n_y-n_y')^2 + (n_z-n_z')^2]) + \beta_{\text{Courbure}} * (\alpha_3 * (k_{\min}-k_{\min}')^2 + \alpha_4 * (k_{\max}-k_{\max}')^2).$$

- Où  $\beta_{\text{Normale}}$  et  $\beta_{\text{Courbure}}$  sont des poids définis dans l'interface.
- Où  $\alpha_1 = 1 / \max$ .
  - $\max$  est le maximum de  $|val_{\min} - val_{\max}|$  des coordonnées des points cibles,  $\alpha_2$  pour les normales.
  - $\alpha_3$  et  $\alpha_4 = 1 / |val_{\min} - val_{\max}|$  pour les courbure principales.
  - $val_{\min}$  est le minimum des minimums de chaque coordonnées des points, des normales ou des courbures principales (resp.  $val_{\max}$ ).

# Résultats du recalage

Testé sur une machine ayant  
core i7, de 2.00GHz et  
8 Go de mémoire vive avec  
une carte graphique NVIDIA Geforce 840M.

# ICP : Interface

The image shows a software window titled "Registration" with several panels for configuring the ICP process:

- Meshes:** Includes dropdown menus for "Source:" and "Target:", a "reinit source and target list" button, and a checkbox "To compute only Matching between Source and Target".
- Points data:** Contains sliders for "weight for normal:" (0.50), "weight for curvature:" (0.50), "remove curv (%):" (0.00), and "weight for colors:" (0.50). It also has checkboxes for "to add the curvature data (min and max)" and "to add the color data".
- For ICP non-deformable (Rigid, Similitude, Affine):** Includes checkboxes for "to add landmarks" and "to choose the number of Source vertice". A text field shows "1 point over 0 = 0 matchings." and a slider for "percentage of the radius for the maximum matchings distance:" (0.30).
- Parameters:** Features a "Load parameters list" button, a checkbox "to store parameters", and a "Save parameters" button.
- Transformation Mode:** Lists options: "RIGID", "SIMILITUDE", "AFFINE (+Similitude)" (checked), "LOCALLY AFFINE (+Affine +Similitude)", and "STEP BY STEP".
- For Locally Affine transformations:** Includes a text field "Percentage of sphere diameter: 5 % of source diameter."
- Errors threshold:** Contains text fields for "Maximum number of iteration: 20" and "Threshold of errors distances: 5 %".
- Display Matching:** Includes checkboxes for "landmarks" (checked) and "vertice". It has a text field "1 matching over 0 = 0 matchings.", a "validate" button, a "width of matching lines: 1.0" field, a "Change vertice color" button, a "Save vertice" button, a "Save the original source vertice" button, a "Draw" button, a checkbox "Draw the distances between Source and Target", and a "Threshold: 0.01" field. A "Display result" button is at the bottom.

At the bottom of the window are "Ok" and "Cancel" buttons.

# ICP : Interface

Sélection des  
maillages source  
et cible

The interface is divided into several sections:

- Meshes:** Contains dropdown menus for 'Source' and 'Target', a 'reinit source and target list' button, and a checkbox 'To compute only Matching between Source and Target'.
- Parameters:** Includes a 'Load parameters list' button, a checkbox 'to store parameters', and a 'Save parameters' button.
- Points data:** Features sliders for 'weight for normal' (0.50), 'weight for curvature' (0.50), and 'weight for colors' (0.50). It also has checkboxes for 'to add the curvature data (min and max)', 'to add the color data', and 'remove curv (%)' (0.00).
- Transformation Mode:** Offers radio button options: 'RIGID', 'SIMILITUDE', 'AFFINE (+Similitude)' (checked), 'LOCALLY AFFINE (+Affine +Similitude)', and 'STEP BY STEP'.
- For Locally Affine transformations:** Includes a field for 'Percentage of sphere diameter' (5) and a label '% of source diameter'.
- Errors threshold:** Contains input fields for 'Maximum number of iteration' (20) and 'Threshold of errors distances' (5 %).
- For ICP non-deformable (Rigid, Similitude, Affine):** Includes checkboxes for 'to add landmarks' and 'to choose the number of Source vertice'. It also has a 'weight for the landmarks' slider (0.00) and a calculation '1 point over 0 = 0 points'. Below this is a section 'For the pairs matching:' with a slider for 'percentage of the radius for the maximum matchings distance' (0.30).

Buttons 'Ok' and 'Cancel' are located at the bottom right.

Sélection  
du  
mode final

Prise en compte  
des landmarks

# ICP : Interface

Donne la dimension des points et leurs poids pour le Kd-tree

Modifie les valeurs extrêmes des courbures principales

Donnée pour filtrer les correspondances

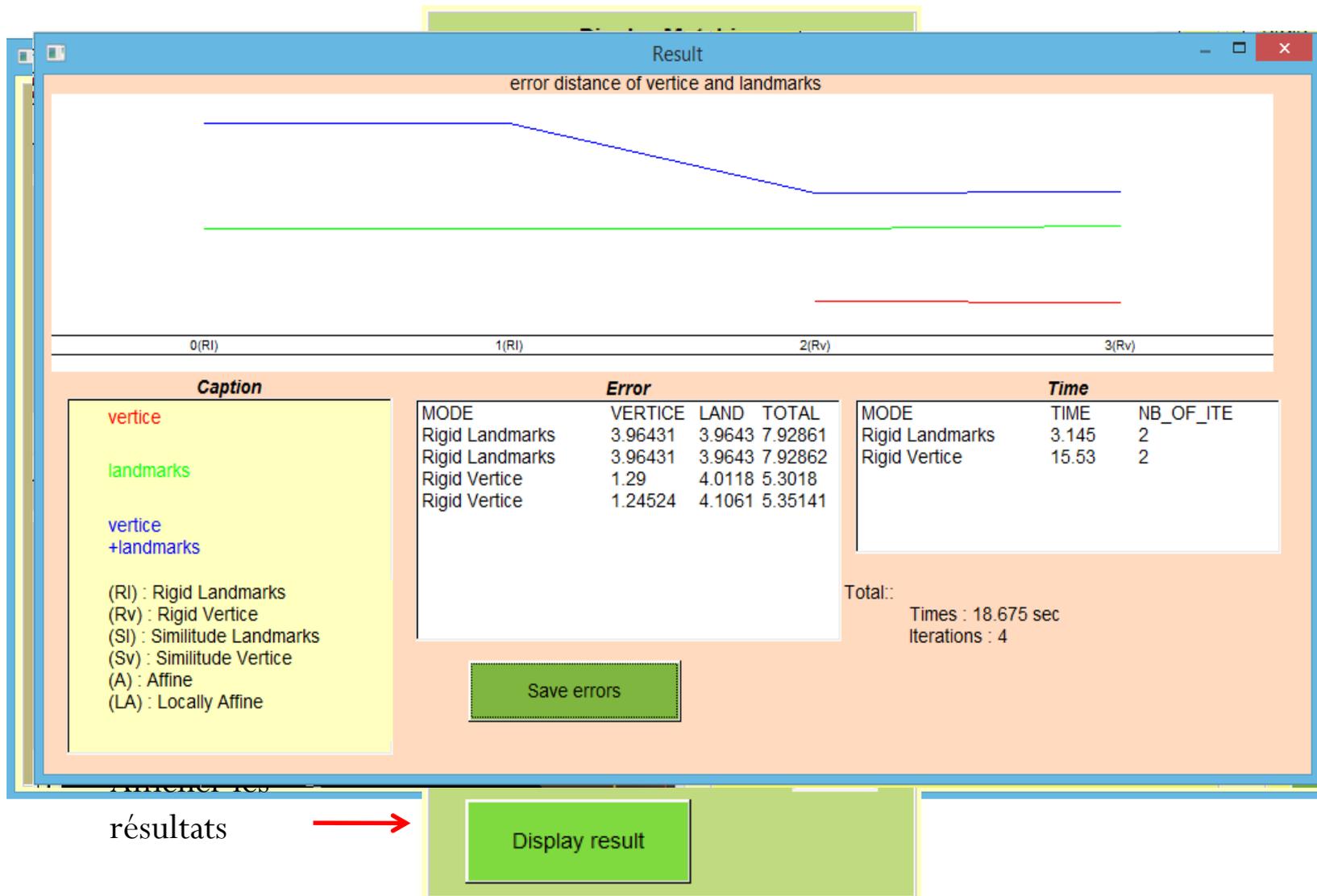
The interface is divided into several sections:

- Meshes:** Contains 'Source' and 'Target' dropdown menus, a 'reinit source and target list' button, and a checkbox 'To compute only Matching between Source and Target'.
- Points data:** Features sliders for 'weight for normal' (0.50), 'weight for curvature' (0.50), and 'weight for colors' (0.50). It includes checkboxes for 'to add the curvature data (min and max)', 'to add the color data', and 'remove curv (%)' (0.00).
- For ICP non-deformable (Rigid, Similitude, Affine):** Includes checkboxes for 'to add landmarks' (weight 0.00) and 'to choose the number of Source vertice'. A field shows '1 point over 0 = 0 points'.
- For the pairs matching:** A slider for 'percentage of the radius for the maximum matchings distance' is set to 0.30.
- Parameters:** Contains a 'Load parameters list' button, a checkbox 'to store parameters', and a 'Save parameters' button.
- Transformation Mode:** Offers options: RIGID, SIMILITUDE, AFFINE (+Similitude) (checked), LOCALLY AFFINE (+Affine +Similitude), and STEP BY STEP.
- For Locally Affine transformations:** A field for 'Percentage of sphere diameter' is set to 5, with the label '% of source diameter'.
- Errors threshold:** Includes 'Maximum number of iteration' (20) and 'Threshold of errors distances' (5 %).
- Buttons:** 'Ok' and 'Cancel' buttons at the bottom.

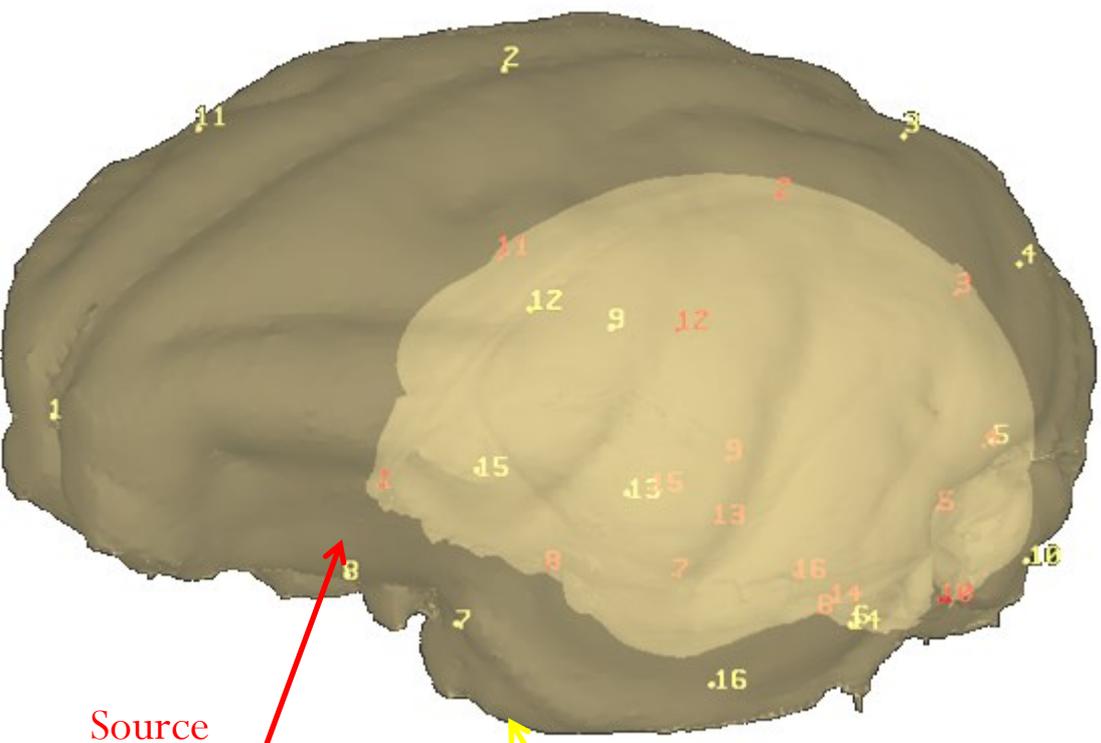
Rayon des sphères( $M_k$ )  
( $R = x\% \times \text{diametre}_{\text{Source}}$ )

Seuil d'arrêt

# Résultats du recalage: Interface



# Résultats du recalage: les données de départ



Source

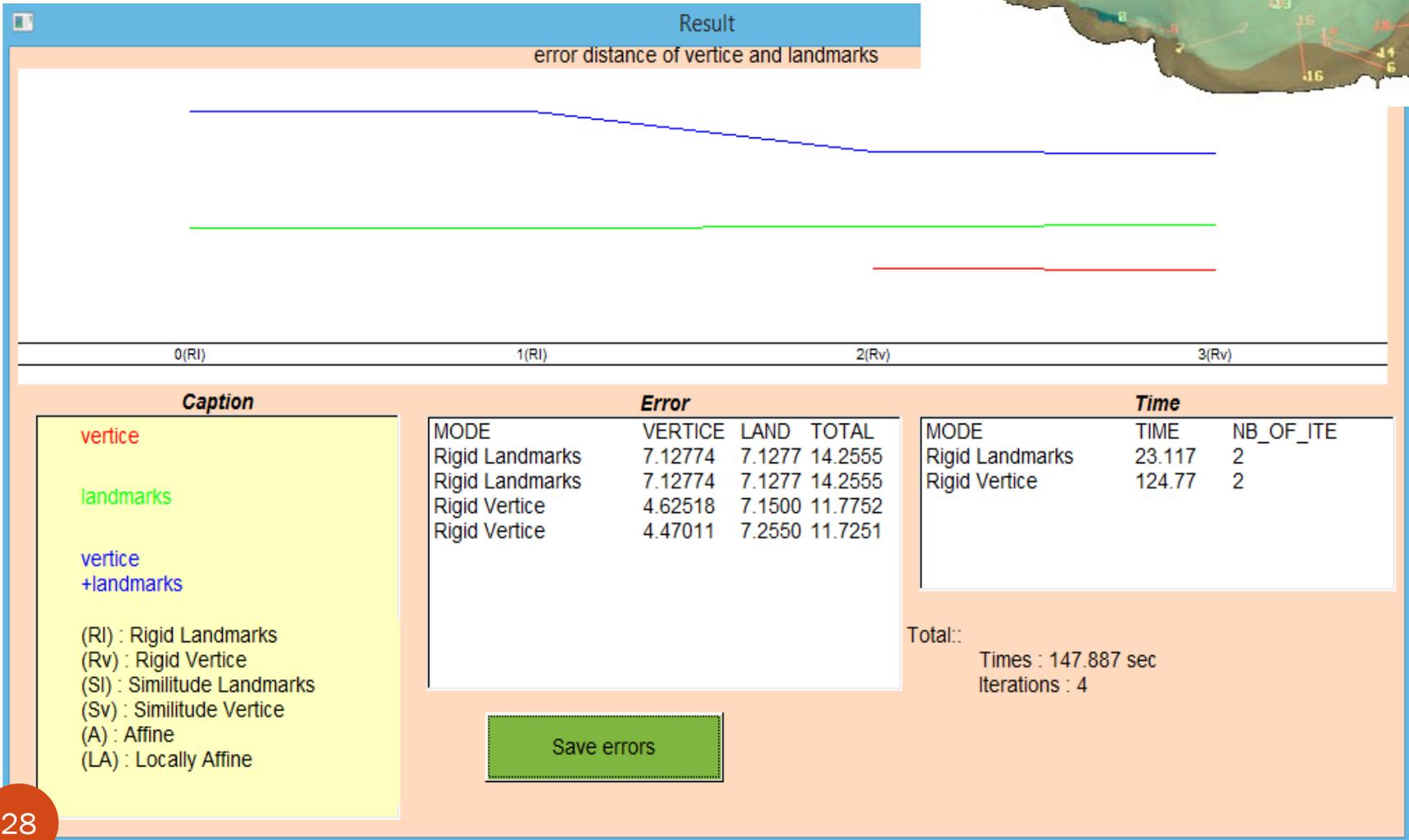
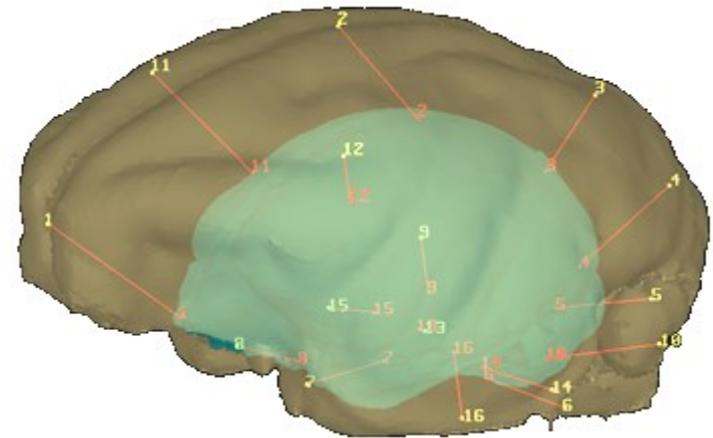
Cible

Cavité endocrânienne d'*Arctocebus*  
contenant 285 665 points  
avec ses 16 landmarks.

Cavité endocrânienne d'*Eulemur fulvus*  
contenant 86 410 points  
avec ses 16 landmarks .

# Résultats du recalage:

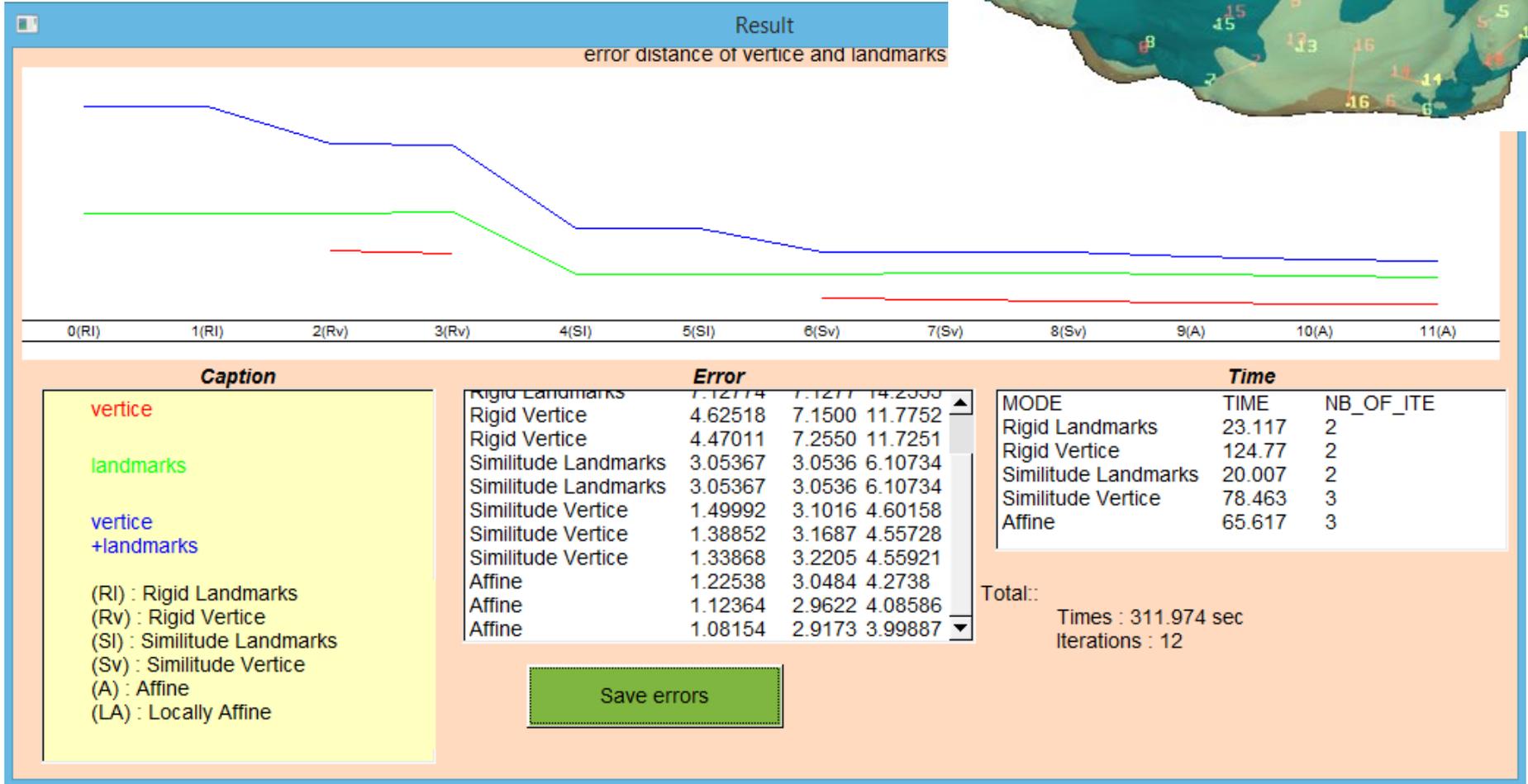
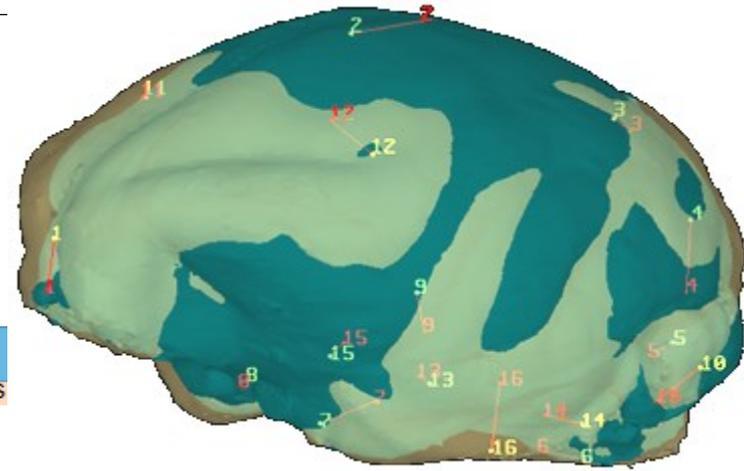
Rigide :



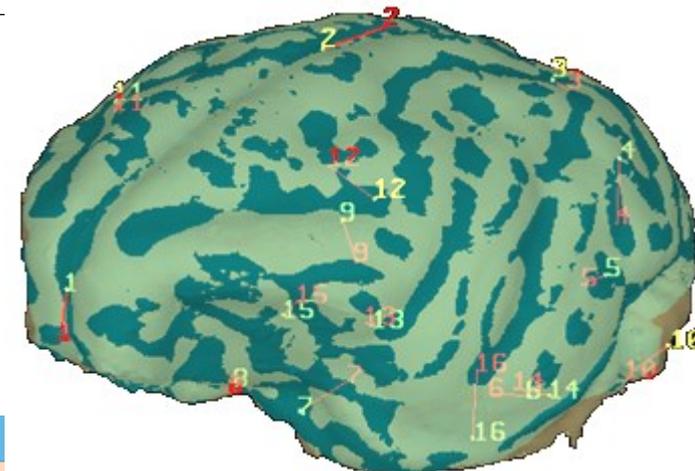


# Résultats du recalage:

Affine :



# Résultats du recalage:



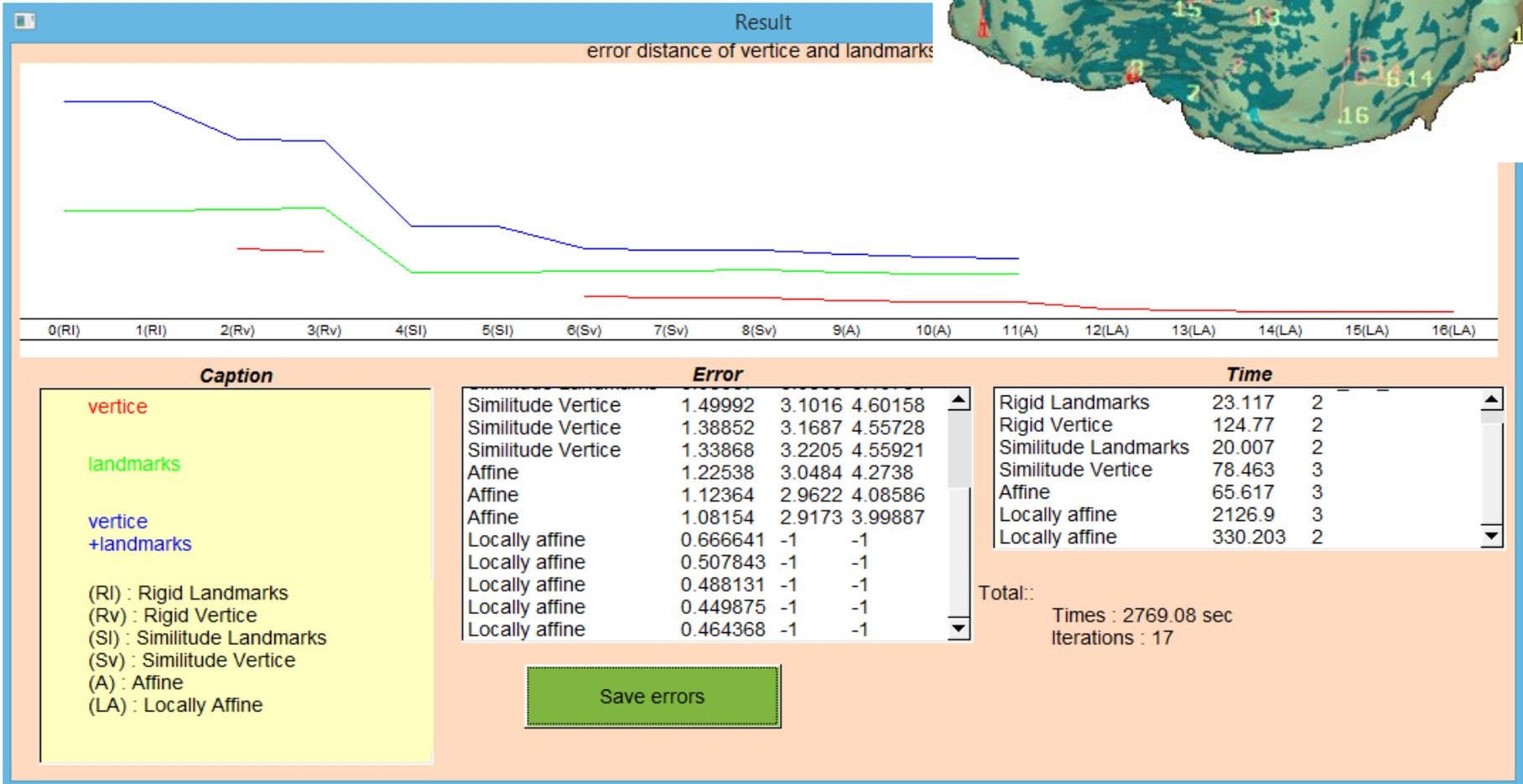
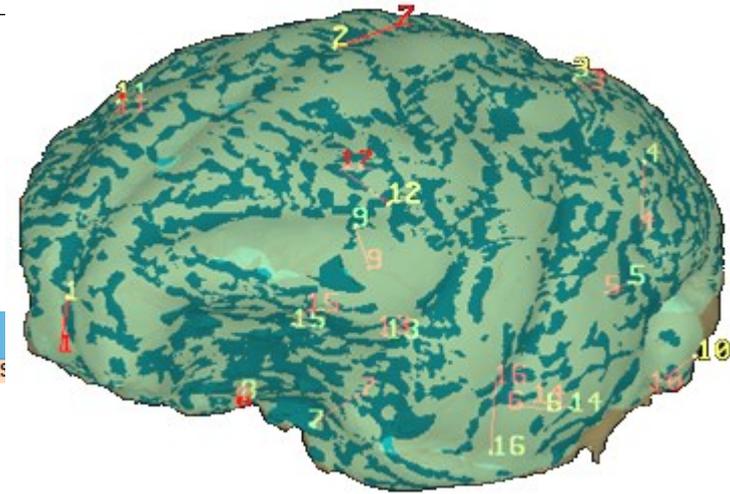
- Localement Affine : Application du pourcentage du diamètre de la source à 5% puis à 2%.

- à 5 %:

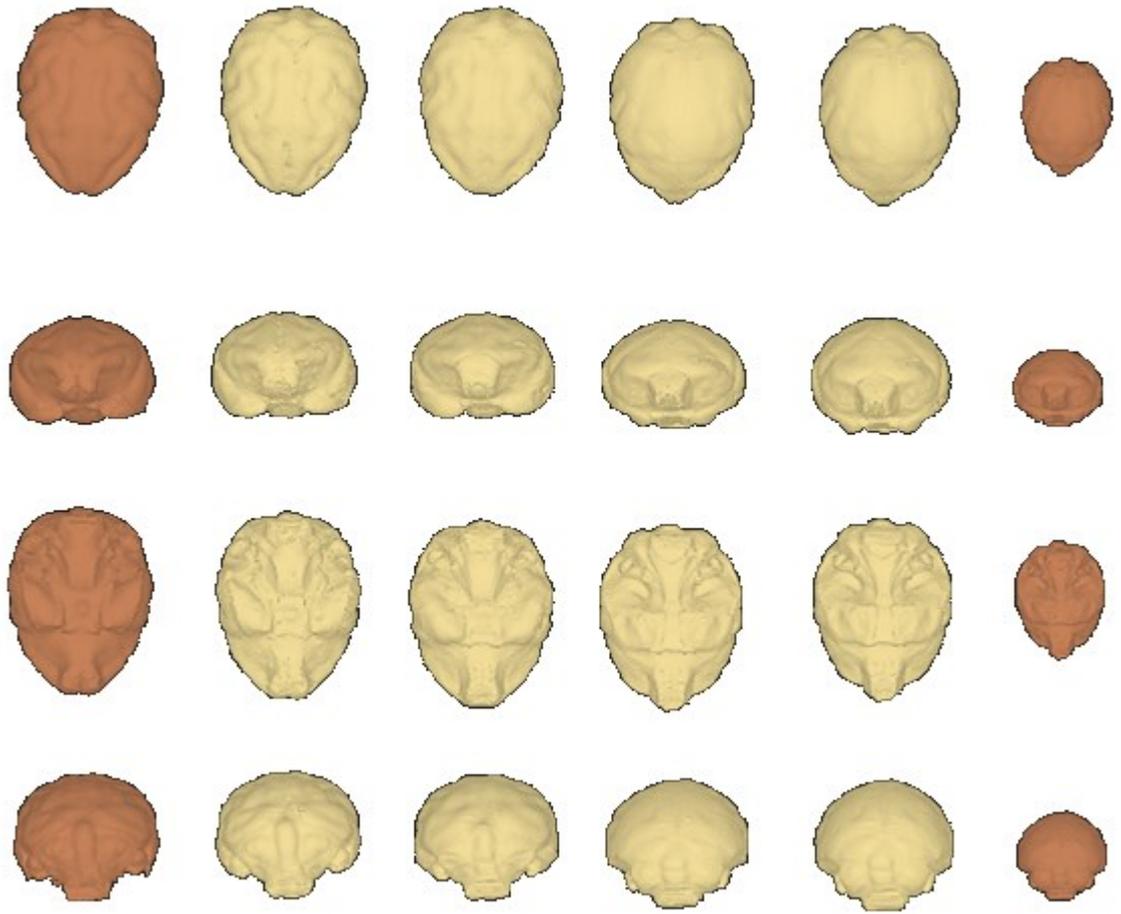


# Résultats du recalage:

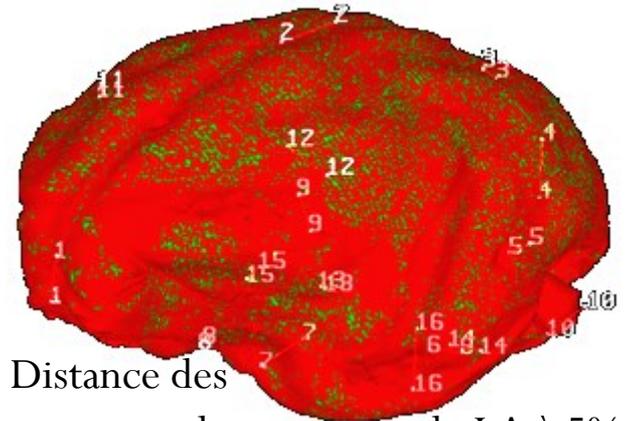
- Puis à 2% :



# Résultats du recalage: Cavité endocrânienne

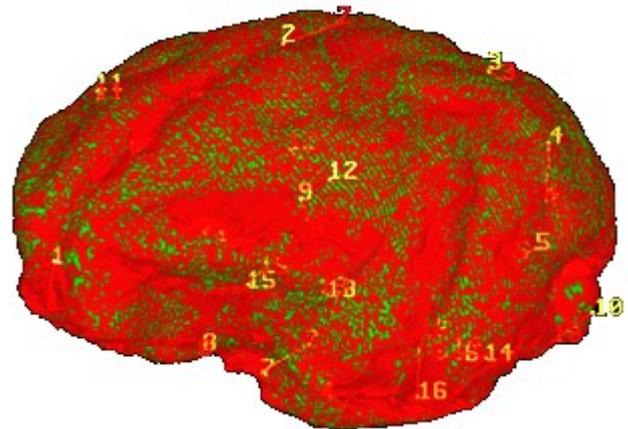


Cible - LA à 5% - LA à 5% - Affine - Similitude - Source  
 puis à 2%



Distance des correspondances en mode LA à 5%.

Distance des correspondances en mode LA à 2%.



En vert,  $\text{dist}(\text{Point}_{\text{Source}}, \text{Point}_{\text{Cible}}) \leq 0.12\text{mm}$   
 sinon en rouge

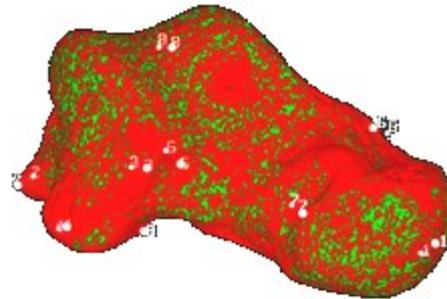
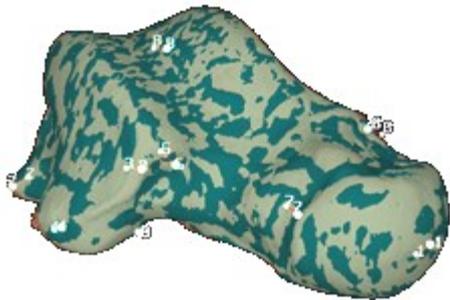
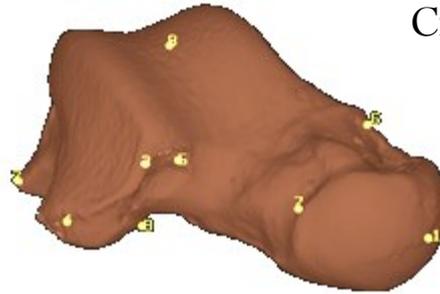
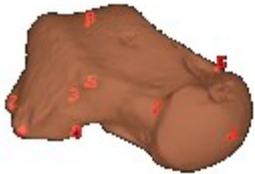
# Résultats du recalage: *Tali* (os du pied)

Source

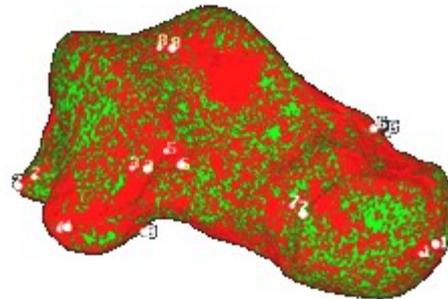
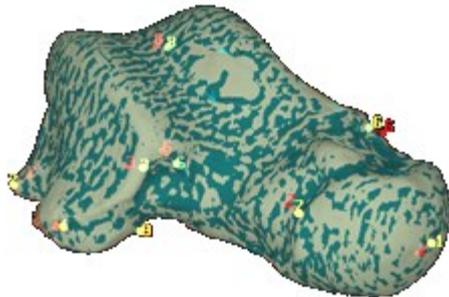
Cible

**Source :** *Talus* d'*Adapis* (entre 38 et 50 millions d'années) avec ses 9 landmarks (130 466 points).

**Cible :** *Talus* de *Lemur Catta* avec ses 9 landmarks (258 663 points).



LA à 5%



LA à 5%  
puis à 2%

Maillage source en bleu  
Maillage Cible en jaune

Distance des correspondances  
(seuil à 0.05mm)

Maillages obtenus

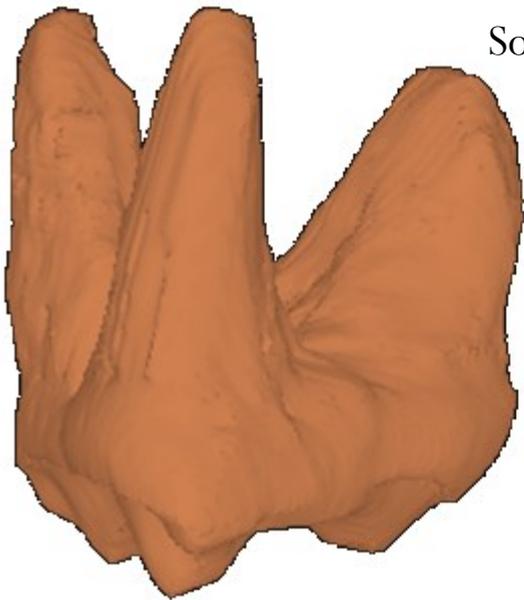
# Résultats du recalage: Molaires sup.

**Cible :** *Phaner* avec  
12 landmarks  
(41 431 points)

**Source :** *Leptadapis*  
avec 12 landmarks  
(102 404 points)



Cible

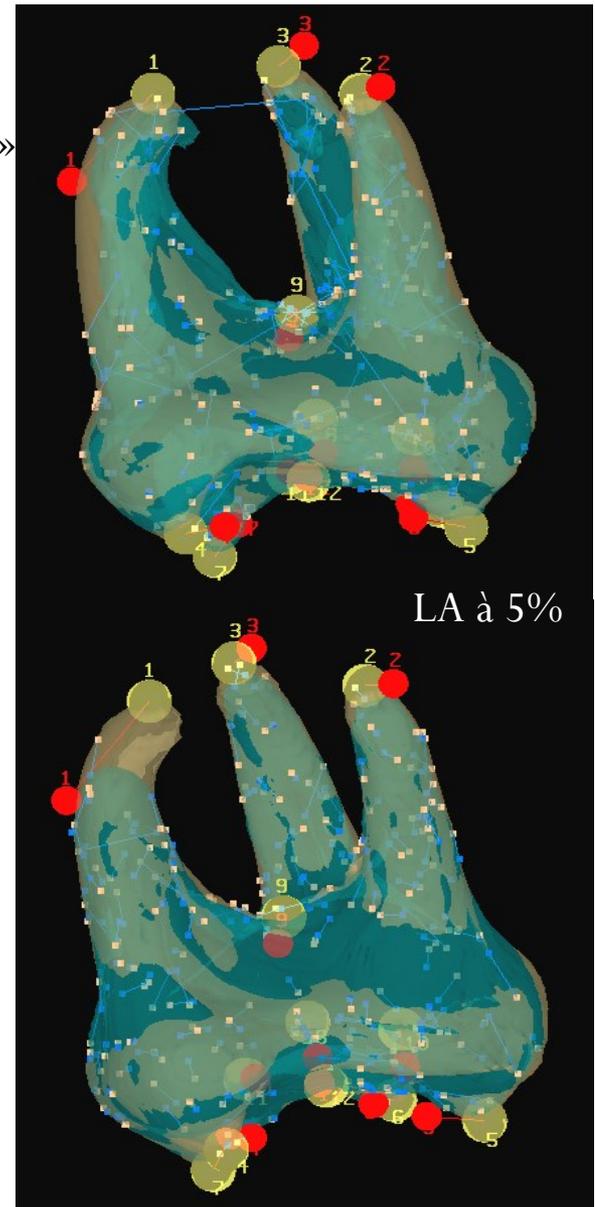


Source

Sans le  
paramètre  
« remove curv »



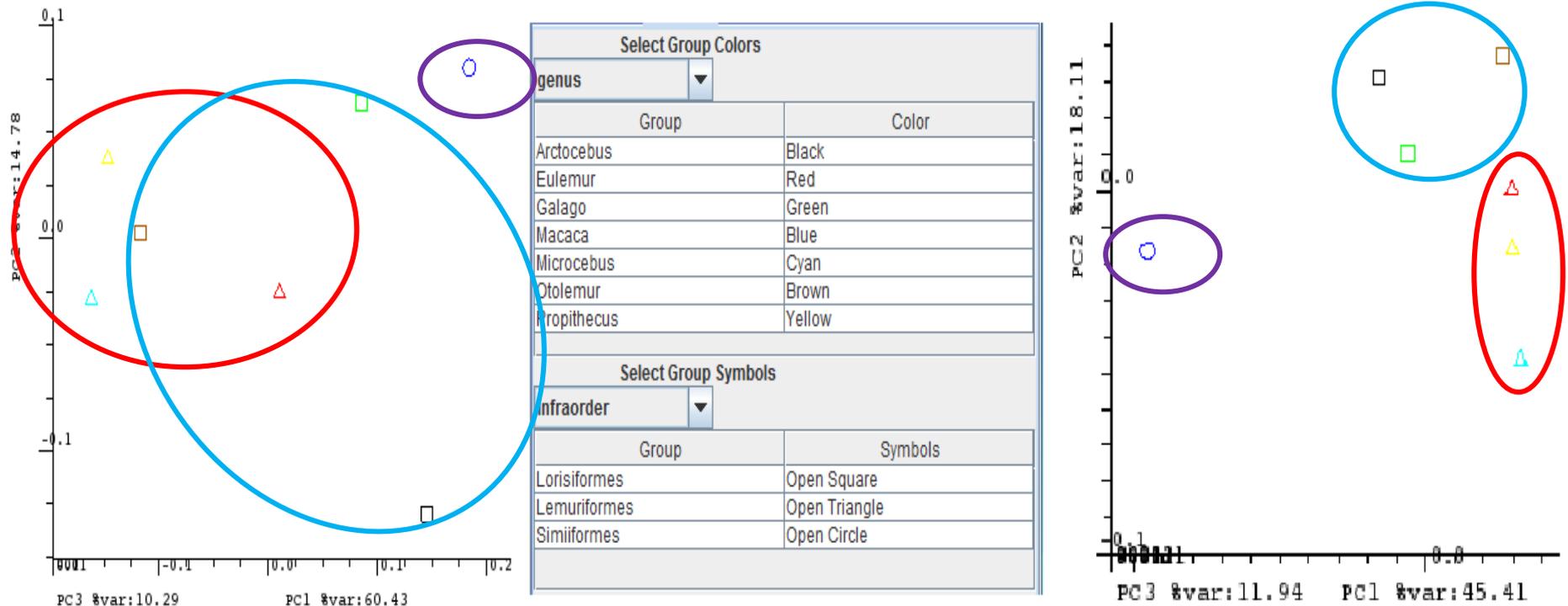
Avec le  
paramètre  
« remove curv »  
à 40%



# Influence du recalage sur l'analyse statistique

Analyse en composante principale (ACP)

7 maillages de primates de cavité endocrânienne

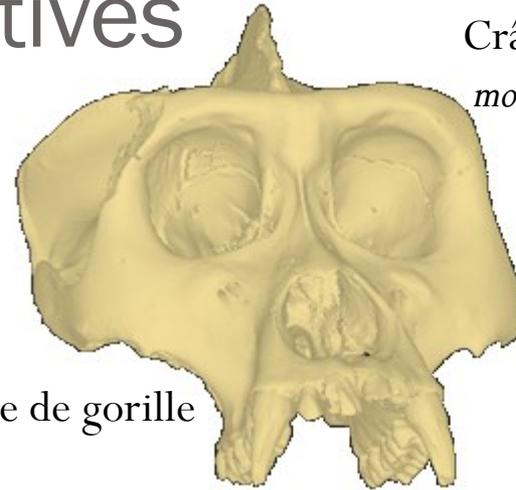


APC avec analyse de 16 landmarks mis par l'expert.

APC avec analyse de 572 appariements sur 285 665 appariements, créés à partir de la méthode recalage automatique.

# Conclusion et perspectives

Crâne d'*Eulemur mongoz* (lémurien)



Crâne de gorille



- Manipulation des maillages :
  - Amélioration nette,
  - Confort pour l'utilisateur.
- Le recalage donne :
  - de bonnes correspondances entre les deux maillages,
  - beaucoup plus de correspondances.
  - Améliorations et perspectives :
    - Tester sur un grand nombre d'échantillons (vertèbres ,dents , *tali*, cavités endocrâniennes)
    - Evaluer les limites de la méthode dans un ensemble d'échantillons en mettant quelques morphologies très éloignées.
    - Ajouter la couleur dans l'étape de l'ICP déformable
    - Utiliser la courbure pour filtrer les correspondances.
- Mise à disposition des codes sources en ligne.

Merci pour votre

attention!



# VBO

```
//construction du buffer
void OBJECT_MESH::Mesh_DrawObj_Building_Buff(){
    // Buffer d'informations de vertex
    glGenBuffers(1, &buf_vertex);
    glBindBuffer(GL_ARRAY_BUFFER, buf_vertex);
    glBufferData(GL_ARRAY_BUFFER, (sizeof(GLfloat) * size_info), ObjectInfoArray, GL_STATIC_DRAW); // si c'est un pointeur noté la taille du tableau directement

    // Buffer d'indices
    glGenBuffers(1, &buf_indice);
    glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, buf_indice);
    glBufferData(GL_ELEMENT_ARRAY_BUFFER, sizeof(GLuint) * size_indice, ObjectIndiceArray, GL_STATIC_DRAW);
}

//on dessine les listes du buffer
void OBJECT_MESH::Mesh_DrawObj_paintGL(){
    glBindBuffer(GL_ARRAY_BUFFER, buf_vertex);
    glVertexPointer(3, GL_FLOAT, ((int)(C_SIZE + 2 * D_SIZE)) * sizeof(float), ((float*)NULL + (C_SIZE + D_SIZE)));
    glNormalPointer(GL_FLOAT, ((int)(C_SIZE + 2 * D_SIZE)) * sizeof(float), ((float*)NULL + (C_SIZE)));
    glColorPointer(4, GL_FLOAT, ((int)(C_SIZE + 2 * D_SIZE)) * sizeof(float), 0);

    glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, buf_indice);

    // Activation d'utilisation des tableaux
    glEnableClientState(GL_VERTEX_ARRAY);
    glEnableClientState(GL_NORMAL_ARRAY);
    glEnableClientState(GL_COLOR_ARRAY);
    // Rendu de notre géométrie
    glDrawElements(GL_TRIANGLES, size_indice, GL_UNSIGNED_INT, 0);

    // Desactivation des tableaux de sommets
    glDisableClientState(GL_COLOR_ARRAY);
    glDisableClientState(GL_NORMAL_ARRAY);
    glDisableClientState(GL_VERTEX_ARRAY);
}
```

# VBO

```
//On récupère les données
void OBJECT_MESH::Mesh_DrawObj_updateBuffer(){
    //VERTEX
    GLfloat * info_vbo = NULL;
    GLuint * indice_vbo = NULL;
    glBindBuffer(GL_ARRAY_BUFFER, buf_vertex);
    info_vbo = (GLfloat *)glMapBuffer(GL_ARRAY_BUFFER, GL_READ_WRITE);

    if (info_vbo == NULL){
        cout << "ERROR, impossible d'accéder au données (info) du vbo!" << endl;
        return;
    }
    memcpy(info_vbo, ObjectInfoArray, size_info * sizeof*ObjectInfoArray); // copie le tableau dans le buffer

    glUnmapBuffer(GL_ARRAY_BUFFER);
    info_vbo = NULL;

    //INDICE
    glBindBuffer(GL_ARRAY_BUFFER, buf_indice);
    indice_vbo = (GLuint *)glMapBuffer(GL_ARRAY_BUFFER, GL_READ_WRITE);

    if (indice_vbo == NULL){
        cout << "ERROR, impossible d'accéder au données (indice) du vbo!" << endl;
        return;
    }
    memcpy(indice_vbo, ObjectIndiceArray, size_indice* sizeof * ObjectIndiceArray);

    glUnmapBuffer(GL_ARRAY_BUFFER);
    indice_vbo = NULL;
}
```