Design and implementation of deformation algorithms for computer assisted orthopedic surgery: application to virtual implant database and preliminary results

S. SAGBO[1,2], G. MARTI[2], C. DI VENUTO[3], C. VANGENOT[3], LP. NOLTE[1], G. ZHENG[1]
[1]MEM Research Center for Orthopaedic Surgery, ISTB, University of Bern, Switzerland

[2]Robotic Systems Laboratory, [3]Database Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Introduction

Fractures around the joints are more common cases in routine trauma surgery where the surgeons have to bend the osteosynthesis plates to fit geometrical boundaries of the specific patient. This is a complicated, time consuming, and technically demanding procedure. The average time for bending a single plate requires as long as twenty minutes. This paper presents the first trial to design and implement the deformation algorithms for computer assisted orthopedic surgery system to assist surgeons in this procedure. Our preliminary results are presented.

Our contribution:

- Select the best appropriate algorithms and adapt them to the surgical procedures
- Perform a first evaluation of their performance

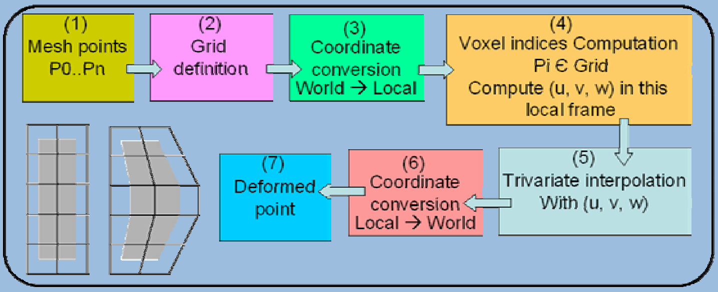
Methods

The different deformation algorithms can be classified in two classes: the geometrical methods (splines-based, FFD, D-FFD, Axial deformation) and the physical methods ("Mass Spring Model", "Finite Elements Model").
We have implemented all axial deformation, E-FFD, D-FDD, and skeleton deformation. Our system main component allows the users to set input parameters via a dedicated graphical user interface (GUI). A visual output with the deformed object is provided as feedback to the users. The critical bending parameters are saved in our implant database [1] via a dedicated interface.

$$\underline{X} = \underline{F}(\underline{x}) \qquad \underline{J}_i(\underline{x}) = \frac{\partial \underline{F}(\underline{x})}{\partial x_i}$$

$$\underline{x} = \underline{x}(u, v); \qquad \underline{n}^{(X)} = \det J\, \underline{\underline{J}}^{-1T} \underline{n}^{(x)}$$

Axial deformation [3, 4]



(1) Mesh points P0..Pn

(2) Grid definition

(3) Coordinate conversion World → Local

(4) Voxel indices Computation Pi ∈ Grid Compute (u, v, w) in this local frame

(5) Trivariate interpolation With (u, v, w)

(6) Coordinate conversion Local → World

(7) Deformed point

Free Form Deformation [2]

Results

The axial deformation algorithms give the expected result. The user can choose the amplification factor, which will result in a bigger or smaller deformation angle. The shear algorithm is much simpler than the bend algorithm and provides the same result for small deformation angles. However for bigger angles it is obvious that the bend algorithm is better than the shear algorithm which is only based on translations. The twist algorithm gives a good result, but it is difficult to control where the implant would be twisted and how to map the amplification factor to the rotation done by the implant. The skeleton-based algorithm was successfully tested for normal and lateral bending and also for torsion-based deformation.

Conclusion

Our laboratory tests show that only a restricted number of the selected algorithms can be adapted and used to simulate the axial, lateral and torsion-based deformations which are the basis of orthopedic plate deformation. So to model realistic deformations, we could use the bend, shear, twist and skeleton-based algorithms. FFD, E-FFD, D-FFD need to be improved (e.g. grid definition and interpolation). The preliminary results are encouraging although improvements and more exhaustive tests with surgeons are required.

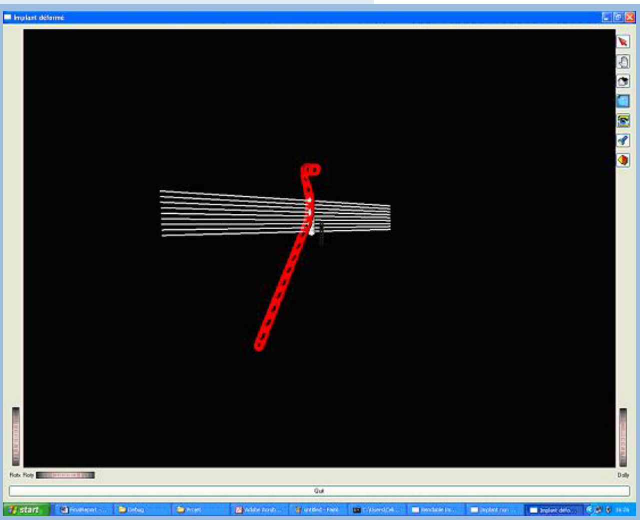


Fig. 1 : The bend result with 10 control Points

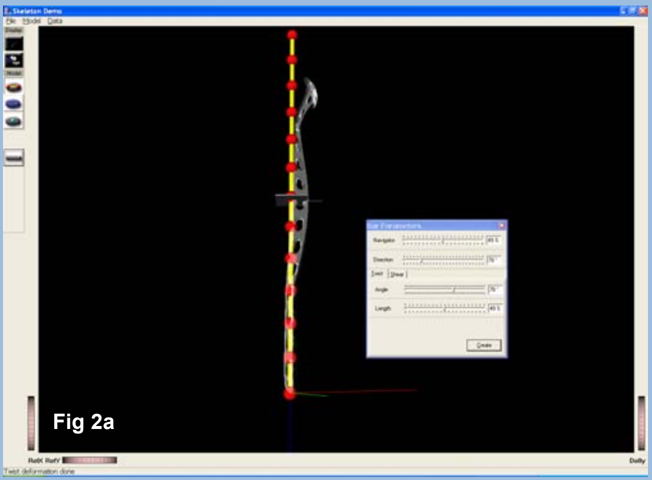


Fig 2a

Fig 2b

Fig 2. Torsion-based deformation with skeleton (a) and axial deformation (b)

References

[1] S. Sagbo, G. Zheng, F. Langlotz, L.P. Nolte. Virtual implants database for Computer assisted surgery: a new system to efficiently access orthopedic implants for planning and navigation. In, Proceedings of CAOS 2004, pages 137-138, 2004, Chicago, Illinois USA
[2] C. Blanc, C. Schlick. Generic Implementation of Free Form Deformation Techniques. In "Graphics Gems V", eds. Academic Press, 1995.
[3] Alan H. Barr. Global and local deformations of solid primitives. Computer Graphics Volume 18, Number 3, July 1984. p.21.
[4] C. Blanc. Generic Implementation of Axial Deformation Techniques. In Graphics Gems, vol 5, p249-256, eds. Academic Press, 1995.

MEM Research Center – ISTB
University of Bern
Staufacherstrasse 78
CH-3014 Bern, Switzerland
+41 31 631 59 55
Serge.Sagbo@memcenter.unibe.ch