

## UN AXIOGRAPHE POUR LA REEDUCATION MANDIBULAIRE UTILISANT LE TRAITEMENT D'IMAGE NUMERIQUE

P. ABELLARD, F. BOUCHARA

Laboratoire MS-LAIAT, Université de Toulon, BP 132, 83957 LA GARDE CEDEX

### RÉSUMÉ

Le papier débute par une présentation du problème à résoudre. La transformation caméra/image réalisée par un modèle de sténopé est définie à partir d'un repère lié à la tête du patient dont on peut prendre ainsi en compte les mouvements. Le traitement des images est ensuite décrit dans ses phases essentielles. On explique enfin, comment sont implémentés les algorithmes de traitement sur une architecture multiprocesseur de type à flux de données qui permet de prendre en compte des contraintes de temps réel. Pour cela, une présentation sommaire des Réseaux de Petri à Flux de Données est effectuée pour permettre de modéliser les calculs.

### 1. INTRODUCTION

La limitation fonctionnelle des mouvements mandibulaires pour des besoins thérapeutiques a souvent été un sujet de controverse en odontologie. En effet, la situation de l'articulation temporo mandibulaire profondément enchassée et la présence des joues et des lèvres rendent difficile l'observation directe des phénomènes [1,2].

Le développement de l'électronique et des ordinateurs a permis de surmonter certaines difficultés et d'avoir une meilleure observation de la fonction manducatrice [3,4,5]. L'axiographe présenté ici, a été développé pour effectuer l'analyse des mouvements complexes des mâchoires afin de contribuer, dans la mesure du possible, à l'amélioration des thérapeutiques.

L'enregistrement des mouvements est réalisé à l'aide de deux caméras numériques à la fréquence de 25 images par seconde. Ni les rapports occlusaux, ni le joint labial ne sont perturbés par la présence de l'appareillage. L'arc facial mandibulaire ne pèse que 60g et la tête du patient est

### ABSTRACT

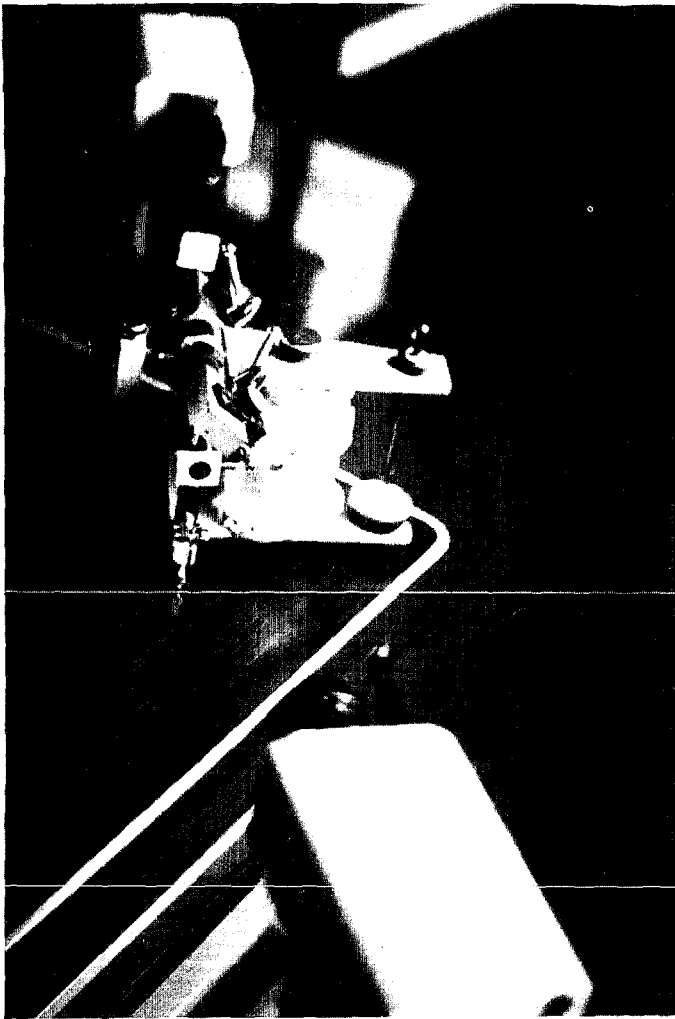
The functional limitation of mandibular movements for therapeutic needs has often been a controversy in odontology. The development of electronics and computers allows to obtain solutions to certain difficulties and to have a better observation of the mastication function. The axiograph presented here has been developed to analyze the complex movements of the mandible in order to contribute, if possible, to the improvement of therapeutics. The movement recording is realized with 2 cameras. Neither occlusal ratios nor the labial joint are disturbed by the presence of the equipment.

libre de toute contrainte. Trois repères contrôlent les mouvements mandibulaires. Les informations obtenues par traitement d'images (lieu de déplacement des condyles, angles d'ouverture...) sont enregistrées dans un fichier pour être ensuite utilisées sous forme de graphiques à l'écran ou sur imprimante.

### 2. LA MANIPULATION

La phase de mise au point de l'axiographe a été développée à partir d'un articulateur classique (photo). La position approximative du condyle est matérialisée par une pastille noire sur un fond blanc afin de l'identifier par un procédé automatique. Cette pastille est fixée sur un arc facial solidaire de la mâchoire inférieure. On prévoit en outre, deux autres pastilles qui vont permettre de définir un repère lié à la tête du patient dont on pourra ainsi prendre en compte les mouvements. Le programme comprend 2 phases

- une phase destinée à repérer les pastilles dans l'image,
- une phase durant laquelle les pastilles seront suivies en temps réel.



### 3. METHODE DE TRAITEMENT

On utilise un modèle sténopé pour décrire la transformation caméra/image. Un point P de l'image 3D forme une image Q dans le plan rétinien (fig 1). On considère la transformation opérée par la caméra comme une transformation perspective parfaite de centre optique O. L'axe optique OK coupe le plan rétinien au point o. La distance Oo est la distance focale de la caméra. Avec le repère 3D de l'espace lié à la caméra (O,I,J,K) et le repère 2D du plan image (o,i,j), nous avons les relations suivantes :

$$\frac{f}{Z} = \frac{x}{X} = \frac{y}{Y} \text{ avec } x = X \frac{f}{Z} \text{ et } y = Y \frac{f}{Z}$$

(X,Y,Z : coordonnées de P dans O,I,J,K)  
(x,y : coordonnées de Q dans o,i,j)

La deuxième transformation effectuée sur l'image est la quantification, c'est à dire la conversion des dimensions en pixels. Elle correspond à un changement d'unités dans le plan image :

$$u = k_x \cdot x \text{ et } v = k_y \cdot y$$

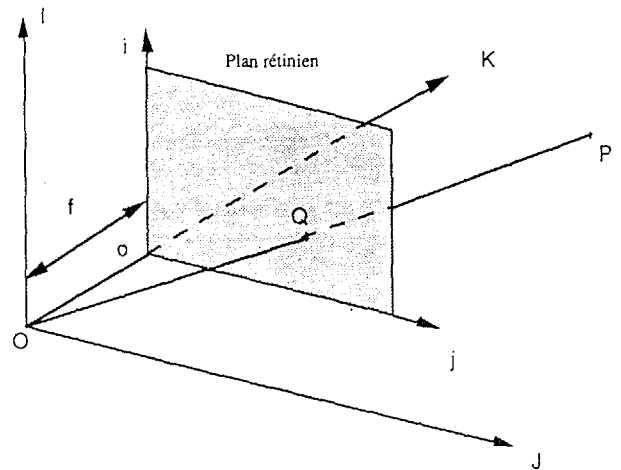


Figure 1 : Transformation perspective.

La quantification s'accompagne généralement d'un phénomène d'écrasement de l'image dû à la différence entre l'échantillonnage effectué par la caméra et celui effectué par le convertisseur A/N de la carte d'acquisition et donc,  $k_x$  et  $k_y$  ne

sont pas égaux :  $k_x = \frac{N'_x}{D_x}$  et  $k_y = \frac{N'_y}{D_y}$  avec  
( $N'_x$ ,  $N'_y$ ) : la résolution de l'affichage,  
( $D_x$ ,  $D_y$ ) : les dimensions du capteur.

Le rapport d'écrasement est  $k = \frac{k_x}{k_y}$

En combinant ces deux effets, on détermine les coordonnées pixels (u,v) dans l'image d'un point de coordonnées (X,Y,Z) dans l'espace :  $u = X \cdot k_u$  et  $v = Y \cdot k_v$  avec :

$$k_u = k \cdot k_y \cdot \frac{f}{Z} \text{ et } k_v = \frac{f}{Z}$$

Comme indiqué, on utilise 2 pastilles pour matérialiser le repère lié à la tête du patient. Ce repère subit les mêmes transformations que précédemment.

Considérons le vecteur  $V_1$  de la figure 2 Dans le repère (o,i,j)  $V_1$  est défini par :  
 $V_1 = V_x \cdot i + V_y \cdot j$  avec  $V_x = ||V_1|| \cdot \cos \alpha$   
et  $V_y = ||V_1|| \cdot \sin \alpha$

Dans le repère (O,I,J), les vecteurs i et j ont les coordonnées suivantes :

$$i = \cos \Phi \cdot I + \sin \Phi \cdot J$$

$$j = \sin \Phi \cdot I + \cos \Phi \cdot J$$

4. ARCHITECTURE DE TRAITEMENT

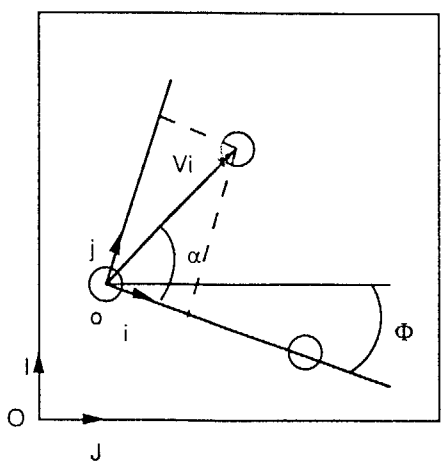


Figure 2 : Définition des repères.

En effectuant un changement de repères, on obtient les nouvelles coordonnées du vecteur  $V_i$  dans  $(O, I, J)$  :

$$V_i = V_x (I \cos \Phi + J \sin \Phi) + V_y (I \sin \Phi + J \cos \Phi)$$

Dans l'espace réel, les coordonnées sont :

$$V_r = T_u (V_x \cos \Phi + V_y \sin \Phi) I_r + T_v (V_x \sin \Phi + V_y \cos \Phi) J_r$$

avec  $T_u = \frac{1}{k_u}$  et  $T_v = \frac{1}{k_v}$

$(O_r, I_r, J_r)$  repère dans le plan analysé.

Soit  $(o_r, i_r, j_r')$  la transformation dans l'espace réel du repère  $(o, i, j)$ . On a :

$$i_r' = \cos \Phi \cdot T_u \cdot I_r + \sin \Phi \cdot T_v \cdot J_r$$

$$j_r' = \sin \Phi \cdot T_u \cdot I_r + \cos \Phi \cdot T_v \cdot J_r$$

$$||i_r'\|| = \cos^2 \Phi \cdot T_u^2 + \sin^2 \Phi \cdot T_v^2$$

$$||j_r'\|| = \sin^2 \Phi \cdot T_u^2 + \cos^2 \Phi \cdot T_v^2$$

En reportant ces valeurs dans l'équation, on obtient :

$$V_r = V_x \cdot i_r' + V_y \cdot j_r'$$

Ce qui permet d'obtenir, après normalisation, la valeur de  $V_r$  :

$$\frac{V_x \cdot i_r}{\cos^2 \Phi \cdot T_u^2 + \sin^2 \Phi \cdot T_v^2} + \frac{V_y \cdot j_r}{\sin^2 \Phi \cdot T_u^2 + \cos^2 \Phi \cdot T_v^2}$$

avec  $(o_r, i_r, j_r)$  le repère normalisé de  $(o_r, i_r', j_r')$ , c'est à dire le repère normalisé défini par les pastilles.

Le problème du calcul parallèle comporte plusieurs aspects différents, souvent liés et qui convergent vers les mêmes buts : augmenter sur le plan matériel et logiciel les performances des systèmes parallèles à évolutions simultanées, et faciliter leur mise en oeuvre. Cependant, ce problème repose sur deux éléments principaux :

- *Modélisation* des programmes parallèles et de leurs structures matérielles de traitement,
- *Choix et conception* d'une architecture parallèle capable d'exécuter des opérations évoluant en même temps.

Le degré de lien entre ces deux éléments joue un rôle important dans la mise en oeuvre d'un processus parallèle car l'architecture matérielle doit s'adapter au modèle de représentation du calcul parallèle et non pas le contraire.

Dans ce but, un **multiprocesseur à Flux de Données** construit à base de processeurs à flots de données dont le fonctionnement est modélisé par Réseaux de Petri à Flux de Données [6] a été étudié et un environnement logiciel spécifique a été créé [7].

Le choix du processeur à flots de données  $\mu$ PD 7281 de NEC a été retenu car ce dernier reflète au niveau du composant la macro-structure de la figure 3 définissant l'architecture à flux de données qui est fondamentalement différente des structures conventionnelles car elle ne comporte pas de processeur central. Il est remplacé par une tranche de modules de traitement élémentaires, unités arithmétiques et logiques, processeurs d'entrée-sorties, etc... Elle ne comporte pas non plus de mémoire centrale à accès aléatoire RAM. Celle-ci est remplacée par une tranche de module mémoires, mémoires d'informations, etc... contenant des codes opératoires, des opérandes, des adresses, etc... Le compteur de programme et les registres de stockage n'existent pas non plus.

Un module de gestion et de commutation de paquets d'opérations (réseau d'arbitrage) permet de relier la sortie d'une tranche de modules mémoires à l'unité de traitement appropriée du système. Chaque unité de traitement envoie un paquet de résultats vers la section de mémoires à travers un réseau de distribution.

Ainsi, dans cette architecture, il n'y a aucun conflit d'accès à un bus ni à une donnée en mémoire. Elle permet donc d'augmenter la vitesse de traitement et de simplifier le développement des



applications. Les Réseaux de Petri à Flux de Données sont utilisés pour la modélisation du fonctionnement du multiprocesseur à flux de données [8]. La figure 4 montre comment ils peuvent remplacer avantageusement pour leur validation, simulation, optimisation, les

graphes de flux : les noeuds correspondent aux places opérateurs, tandis que la notion d'arc est prise au sens large de liaison inter-opérateurs par une place variable. Dans le langage fonctionnel d'assemblage utilisé, le programme source décrit la fonction à exécuter et ne spécifie pas toutes les étapes comme pour les assembleurs habituels.

5. CONCLUSION

Ainsi, nous pouvons plus facilement aborder la véritable physiologie de l'appareil manducateur qu'est l'étude des mouvements fonctionnels dans leur champ d'expression quotidien constitué essentiellement par la mastication, la déglutition, la phonation et la respiration par un enregistrement beaucoup plus aisé de la cinématique rendue possible grâce au traitement des images.

**MOTS CLES** : Traitement d'images, stéréoscopie, axiographe, rééducation mandibulaire.

BIBLIOGRAPHIE

[1] Gaspard M : L'appareil manducateur et la manducation : anatomie descriptive, ontogénèse et phylogénèse de la mandibule humaine. Edit. J. Prélat, Paris 1979

[2] Yaeger JA : Mandibular path in the grinding phase of mastication. *J. Prosth. Dent* n°39, pp 569-573, 1978.

[3] Duret F, Toubol JP, Jordan F, Georget C : La lecture des mouvements mandibulaires par un système opto-électronique. *Quest. Odonto-Stomat.* n°438, pp 21-29, 1988.

[4] Hobo S, Hayakawa H, Takayama H, Iwata T, Hawazu H : An experimental electronic recording system for the motion of the condyle in three dimensions, *J. Gnathology* n°2.1, 1983.

[5] Mongini F, Tempia-Valenta G, Conserva E : Habitual mastication and dysfunction A computer based analysis, *J. Prosth. Dent.* n°4, pp 484-494, 1989.

[6] Almhana J : Modélisation par Réseaux de Petri à Flux de Données. Application à la synthèse de l'opérateur de Riccati rapide. *Thèse d'Etat*, Marseille, 1983.

[7] Abellard P, Nolibe G, Razafindrakoto N A Data Flow Architecture for Parallel Computation. *International Conference PARBASE 90*, Miami IEEE Computer Press, 1990

[8] Abellard P, Salgon E, Richon C, Balmat JF : PFLUX, A software tool for Data Flow Petri Net Design. *The Petri Net Newsletter*, n°36, pp 7-13, August 1990.

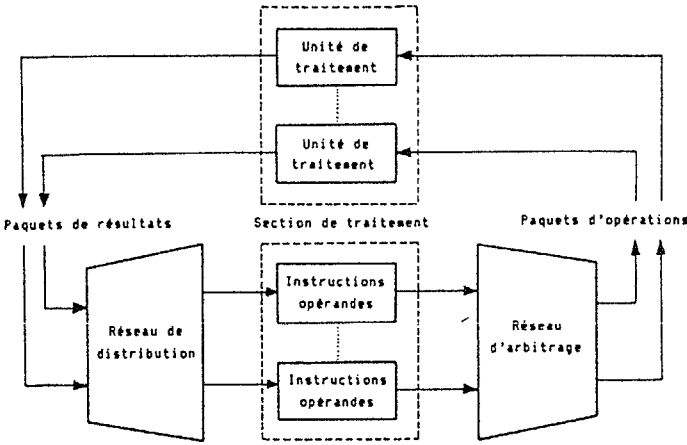
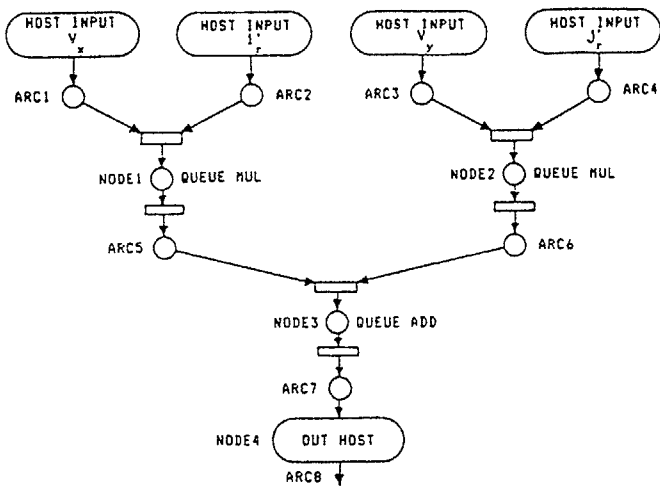


Figure 3 : Architecture à flux de données.



```

1. EQUATE HOST=0;
2. MODULE EXONE=1;
3. INPUT ARC1,ARC2,ARC3,ARC4;
4. OUTPUT ARCB;
5. LINK ARCS=NODE1 (ARC1,ARC2);
6. LINK ARCA=NODE2 (ARC3,ARC4);
7. LINK ARC7=NODE3 (ARC5,ARC6);
8. LINK ARCB=NODE4 (ARC7, );
9. FUNCTION NODE1=MUL,QUEUE (QUE1,1);
10. FUNCTION NODE2=MUL,QUEUE (QUE2,1);
11. FUNCTION NODE3=ADD,QUEUE (QUE3,1);
12. FUNCTION NODE4=OUT1 (HOST,0);
13. MEMORY QUE1=AREA (1);
14. MEMORY QUE2=AREA (1);
15. MEMORY QUE3=AREA (1);
16. START;
17. DATA EXEC (EXONE,ARC1);
18. DATA EXEC (EXONE,ARC2);
19. DATA EXEC (EXONE,ARC3);
20. DATA EXEC (EXONE,ARC4);
21. END.
    
```

Figure 4 : Réseau de Petri à Flux de Données et programmation sur processeur μPD 7281 du calcul  $V_r = V_x \cdot i_r' + V_y \cdot j_r'$