Groupe Vision par Ordinateur

Mémoire de Licence

Rx-Cervicales, Système d'Analyse pour la Radiologie Cervicale Fonctionnelle

Daniel Doubrovkine
doubrov5@cuimail.unige.ch
sous la direction du Prof. Thierry Pun

Université de Genève
Département d'Informatique, Faculté de Sciences
24, rue Général-Dufour
1211 Genève 4

1 avril 1999

Résumé

Dans leur travail quotidien avec des patients, des chirurgien et des neurologues très spécialisés sont amenés à examiner des centaines d'images radiographiées et à détecter des dizaines d'anomalies telles que des déplacements vertébraux. Ce travail minutieux nécessite un professionnalisme sans égal et une concentration extrême. Les cas étudiés ne présentent que très rarement des problèmes visibles à l'œil nu et des mesures particulièrement précises décident toujours d'un besoin d'intervention du médecin.

Ainsi, les hôpitaux utilisent des techniques d'avant-garde en matière d'informatique et ont poussé le développement de logiciels aidant le médecin dans son travail avec les patients. Désormais, le domaine de l'imagerie médicale et plus particulièrement celui du traitement d'images radiographiées propose un vaste champ d'application pour des travaux de recherche.

Dans le présent document nous nous intéressons aux radiographies de patients en traitement à l'Hôpital Cantonal Universitaire de Genève et plus particulièrement de la Clinique de Rééducation. Nous présentons tout d'abord une base logicielle pour des applications dans le domaine d'imagerie médicale suivie d'une automatisation de mesures sur les angles intervertébraux.

De nombreux groupes de recherche se sont intéressés aux coupes effectuées par des scanners IRM et ont produit des logiciels d'analyse poussés. Les aspects comme la simulation en trois dimensions et la médecine à distance sont largement explorés. En revanche, nous offrons une approche plus axée sur les mesures de distances et d'angles. Ainsi, Rx-Cervicales, le logiciel développé dans le cadre de ce travail offre principalement une base vectorielle mais n'omet toutefois point l'aspect traitement d'images grâce à de nombreux filtres. Rx-Cervicales permet également de superposer des images, d'en modifier la taille et d'effectuer des déplacements et des rotations libres.

Par la suite, nous étudions ici la reconnaissance de contours sur des calques de radiographies et permettons au médecin de mesurer des angles entre deux clichés de flexion et d'extension de la colonne cervicale ou lombaire du même patient puis de comparer ces résultats avec des valeurs d'individus sains. Ainsi, des anomalies peuvent être détectées par l'ordinateur en quelques secondes. Notons que notre produit est directement utilisé par la Clinique de Rééducation de l'HCUG.

En conclusion, soulignons que malgré le nombre important de groupes de recherche et d'institutions travaillant dans ce domaine, il n'existe pas à notre connaissance, de logiciels semblables à celui proposé dans le présent travail de recherche. Donc, nous fournissons un outil, muni d'une première application, qui montre l'efficacité de l'approche utilisée et laisse entrevoir de larges perspectives d'avenir.

1 Introduction ..... page 4
1.1 Préambule ..... page 4
1.2 Contexte et Problématique ..... page 4
1.3 Outils Existants ..... page 5
1.4 Logiciels Comparables ..... page 5
2 Objectifs ..... page 7
2.1 Buts ..... page 7
2.2 Cahier de Charges ..... page 8
2.2.1 Système, Support et Programmation ..... page 8
2.2.1 Général ..... page 8
2.2.2 Outils Vectoriels ..... page 8
2.2.3 Images et Traitements ..... page 8
3 Solutions et Implémentation ..... page 10
3.1 Architecture ..... page 10
3.1.1 Base Vectorielle ..... page 10
3.1.2 Support Bitmaps ..... page 10
3.1.3 Interface Utilisateur ..... page 10
3.1.4 Automatisations Spécifiques à Rx ..... page 10
3.2 Implémentation ..... page 11
3.2.1 Base Vectorielle ..... page 11
3.2.2 Support Bitmaps ..... page 12
3.2.3 Opérateurs Avancés sur les Bitmaps ..... page 14
3.2.4 Interface Utilisateur ..... page 17
3.2.5 Automatisations Angulaires ..... page 18
4 Résultats et Conclusion ..... page 21
4.1 Résultats ..... page 21
4.1.1 Contexte ..... page 21
4.1.2 Clichés et Calques ..... page 21
4.1.3 Automatisations Angulaires ..... page 22
4.1.4 Résultats ..... page 22
4.2 Evolutions Futures ..... page 23
4.3 Conclusion ..... page 24
5 Références, Bibliographie, Annexes ..... page 25
5.1 Références, Bibliographie ..... page 25
5.1.1 Imagerie Numérique ..... page 25
5.1.2 Programmation ..... page 25
5.1.3 Autres ..... page 25
5.2 Remerciements ..... page 25

Chapitre 1

1 Introduction

1.1 Préambule

La Suisse a compté un triste score de 14'000 blessés et 300 tués suite à de divers accidents de voiture en 1998 [1]. Des traumatismes de colonne vertébrale ou cervicale résultants d'accrochages sur les routes représentent environ 200 cas par année. La clinique de Rééducation de l'Hôpital Cantonal Universitaire Genevois accueille près de 40% de ces patients. L'équipe de dix neurochirurgiens pratique plus de 25 interventions sur des patients au niveau de la colonne vertébrale, cervicale ou lombaire chaque année[2].

Afin de déterminer si une intervention chirurgicale est nécessaire ou pour suivre l'évolution des traumatismes vertébraux de nombreuses radiographies sont régulièrement prises. L'évaluation de ces dernières est à la charge des médecins et constitue un temps non négligeable de travail. Pour les colonnes cervicales en particulier il s'agit de radiographier le cou dans les deux positions de flexion et d'extension, puis de comparer les deux clichés afin de déterminer d'éventuels déplacements de vertèbres.

Le projet Rx-Cervicales à été proposé par Dr. Paul Cohen (Paul.Cohen@hcuge.ch), chef de clinique adjoint de la Clinique de Rééducation de l'Hôpital Cantonal de Genève en collaboration avec le Département d'Informatique de l'Université de Genève dans le but d'informatiser certaines opérations répétitives que le médecin effectue pour chaque patient.

Le présent travail de licence est effectué sous la direction du Prof. Thierry Pun (Thierry.Pun@cui.unige.ch) du groupe Vision par Ordinateur.

Le logiciel Rx-Cervicales, développé dans le cadre de ce travail et décrit dans le présent rapport est employé à l'Hôpital Cantonal de Genève. Son utilisation dans d'autres hôpitaux suisses ou étrangers est possible et même souhaitable. Notons également que le médecin est le seul apte à juger d'un besoin de traitement, le présent travail se limite à la création d'un outil informatique d'aide à la mesure d'angles intervertébraux.

1.2 Contexte et Problématique

Le projet vise une automatisation (informatisation) des mesures d'angles intervertébraux sur des radiographies fonctionnelles de la colonne vertébrale. En particulier, sur des radiographies de la colonne cervicale (fig. 1.1). Cette dernière qui ne montre que le cliché standard de profil, représente le point de départ de l'analyse.

fig. 1.1

Les angles mesurés correspondent au déplacement d'un corps vertébral par rapport au corps vertébral sous-jacent entre les positions d'extension et de flexion de la colonne cervicale (fig. 1.2). La figure 1.2 constitue la seconde étape où les corps vertébraux ont été reportés sur un transparent afin de pouvoir superposer les 2 images.

C'est en superposant les 2 transparents par rapport à chaque corps vertébral que l'on peut déterminer l'angle entre 2 vertèbres. Un tel angle est illustré également sur la fig. 1.2. Ces mesures sont ensuite comparées à des valeurs normales déterminées chez des sujets sains.

Cette comparaison peut ensuite mettre en évidence une augmentation ou une diminution de la mobilité d'un ou de plusieurs niveaux vertébraux. Cette automatisation permettrait un gain de temps considérable, car la méthode manuelle demande au minimum 15 minutes par couple de clichés.

fig. 1.2

1.3 Outils Existants

La spécificité de logiciels tel que Rx-Cervicales et les prix élevés de solutions existantes poussent la création de départements spécialisés tel que le Centre d'Imagerie Médicale de l'HCUG [3] et les développements en interne de programmes. Malgré un champ d'applications informatiques dans le domaine de traitement d'images médicales immense, le manque de collaboration entre hôpitaux étrangers ou même suisses induit des efforts similaires de part et d'autre.

A ce jour, l'Hôpital Cantonal de Genève ne possède pas de logiciels semblables au projet Rx Cervicales. Néanmoins, des utilitaires couvrant une partie des objectifs de ce projet sont bien présents dans le domaine médical. Il existe également de nombreuses applications de traitement d'images radiographiées mais à ma connaissance aucun logiciel ne possède une gamme suffisamment large d'outils vectorielle pour les mesures de longueurs et d'angles, pouvant traiter des calques ou des images radiographiées à cet effet.

Parmi les logiciels connus citons Navimetric Tomometric 2D [4] et Osiris [5]. Ce dernier sera présenté ci-dessous.

1.4 Logiciels Comparables

L'Hôpital Cantonal de Genève utilise actuellement un logiciel appelé Osiris (fig 1.3), développé à l'Unité d'Imagerie Numérique de l'HCUG dans le cadre du projet PACS [6]. Osiris est un logiciel de manipulation et d'analyse d'images médicales et possède une gamme intéressante d'outils qui s'appliquent en partie au problèmes posés par ce projet de recherche.

fig. 1.3

Osiris prévoit en théorie des extensions faciles à programmer - ainsi il aurait été possible d'intégrer Rx Cervicales en tant que plugin utilisant les fonctions de base d'Osiris. Certaines limitations et certains aspects que Osiris ne couvre pas et décris ci-dessous poussent à développer un logiciel nouveau et de ne pas se limiter à une solution minimale purement fonctionnelle.

- Osiris n'est qu'à moitié vectoriel: les images sont traitées séparément et il n'est pas possible d'intégrer plusieurs images et de les superposer en transparence. En revanche, Osiris possède une vaste palette de traitement d'images.
- Osiris est dépourvu d'outils de rotation d'images sur des angles quelconques.
- Les outils de tracé et de calcul d'angles sont limités. En occurrence, Osiris ne permet pas de calculer des intersections d'angles entre segments aléatoires.
- L'architecture d'Osiris est SDI (Single Document Interface). Plusieurs documents ne peuvent être ouverts à la fois et de multiples clichés ne peuvent donc être comparés.
- L'impression dans Osiris n'est que très sommaire et la prévisualisation inexistante. En revanche, Osiris possède un système de gestion d'images propriétaire avec annotations standard sur le patient et la nature des problèmes.
- Certains outils, notamment les annotations, sont très minimaux et peu pratiques à utiliser.

Malgré ces défauts, Osiris apporte de nombreuses facilités pour les médecins de l'hôpital et est connu dans le monde entier.

Chapitre 2

2 Objectifs

2.1 Buts

En pratique, on doit envisager 2 logiciels différents, en fonction de l'équipement du médecin utilisateur:

- Si le médecin ne dispose que d'un scanner standard, le logiciel doit pouvoir travailler sur les images décalquées (fig 1.2).
- Si ce dernier dispose d'un scanner pour transparent il devrait pouvoir travailler directement depuis les clichés radiologiques (fig 1.1).

Le présent travail de licence se limitera à traiter la première de ces deux situations (calques). Le projet se décompose en trois étapes décrites ci-dessous. L'étape 4 est présentée en tant que perspective d'évolution et devra être prise en considération tout au long du développement.

1. Le but de la première phase du développement sera de créer une base logiciel à la mesure d'angles, au tracé de droites, de segments joints, parallèles ou perpendiculaires. Ceci sera indépendant du problème posé et pourra être réutilisé dans beaucoup d'autres applications et dans de nombreuses évolutions possibles du logiciel. Il faudra également donner à l'utilisateur la possibilité de sauvegarder des documents et de les recharger ou distribuer, les imprimer et aussi permettre à travailler sur plusieurs documents en même temps. Il faudra pouvoir mesurer des distances et définir des échelles de mesure.

2. La seconde phase du présent travail sera de proposer une solution semi-automatique pour la mesure d'angles entre les calques de flexion et d'extension. L'utilisateur devra placer des points de contrôle qui serviront au logiciel pour calculer les différences d'angles.

3. Après la troisième phase d'évolution, le logiciel pourra reconnaître les contours des calques et placer les points de contrôle automatiquement. Ceci terminera le présent travail de licence et finalisera les outils essentiels nécessaires pour une étape de développement ultérieur.

4. La quatrième phase, à savoir un développement vers une version complexe traitant les IRM, à et une reconnaissance complète d'une image radiographiée reste possible. Cette éventualité sera considérée tout au long du travail afin de ne pas défavoriser une telle évolution. Ceci pourra faire l'objet d'un autre sujet de travail.

2.2 Cahier de Charges

2.2.1 Système, Support et Programmation

- Le logiciel aura pour plateforme Microsoft Windows 95/98 et Windows NT. Ce choix est déterminé par la disponibilité du matériel à l'Hôpital Cantonal de Genève.
- Le logiciel sera écrit en C++ avec un soucis permanent d'un éventuel port sur des plateformes UNIX.

2.2.1 Général

- Interface MDI (multiple document interface) utilisant l'architecture Document-Vue afin de pouvoir travailler sur plusieurs documents simultanément, traiter plusieurs affichages, etc.
- Impression papier et visualisation WYSIWYG, scrolling, tailles de documents standards (A4, US Letter, etc.).
- Implémentation de Windows OLE (object linking and embedding), copier-coller d'images d'autres applications Windows.
- Sauvegarde et chargement de documents.
- Importations de formats standards d'image (au minimum Windows BMP).

2.2.2 Outils Vectoriels

- Tracé vectoriel de droites simples, de segments liés ainsi que d'autres éléments géométriques.
- Mesure automatique d'angles entre toutes les intersections.
- Déplacement, modification et suppression de chaque élément d'une manière vectorielle, couleurs pour plus de clarté, etc.
- Annotations textuelles en tant que éléments indépendants, édition, suppression, etc.
- Mesure de longueurs avec un segment ou une suite de segments.
- Possibilité de définir une échelle de mesure.
- Traitements vectoriels d'images tels que la modification de la taille et la rotation.

2.2.3 Images et Traitements

- Traitements simples d'image tels que la conversions en N/B ou niveaux de gris, inversion.
- Identification de formes guidées, reconnaissance de contours manuelle et création de rapports de différences d'angles.

- Traitements de convolution: adoucissement (blur), rehaussement de contours (sharpen) et d'autres filtres utilisant une matrice de convolution.
- Détection et traitement de contours: amincissement, opérateurs de Sobel, Roberts et opérateur Laplacien.
- Affichage d'informations sur l'image: légendes comprenant la taille et la profondeur, histogrammes de niveaux de gris et loi cumulative.

Chapitre 3

3 Solutions et Implémentation

3.1 Architecture

L'architecture du logiciel Rx Cervicales se divise en plusieurs modules distincts afin d'assurer un maximum d'évolution et permettre une portabilité vers un système UNIX avec un minimum d'efforts.

3.1.1 Base Vectorielle

Le premier module est une série de classes dérivées plus ou moins indépendantes du modèle de l'interface qui implémentent chacune les fonctions particulières vectorielles. Un tel objet de base possède systématiquement une interface (un manager) capable d'organiser une suite d'éléments homogènes, de les sérialiser et s'occupant également de transmettre les messages tels que les déplacements de souris ou les touches du clavier.

Ce module comprend toutes les implémentations de segments et de vecteurs, des annotations textuelles et des manipulations vectorielles sur les images (rotation,élargissement ou réduction).

3.1.2 Support Bitmaps

Les opérations sur le contenu des images (qui sont des éléments semi-vectoriels dans Rx) sont définies par le second module. Les conversions de couleurs, le traitement des informations telles que les histogrammes, les accès aux pixels individuels, les opérateurs de convolution sont tous assurés par ce dernier.

3.1.3 Interface Utilisateur

Le troisième module est une interface utilisateur respectant à la lettre le modèle Document/Vue - ce choix se motive par l'implémentation des MFC (Microsoft Foundation Classes). Le modèle Document/Vue défini pour la première fois dans Smalltalk et popularisé par les MFC (Microsoft Foundation Classes) est un principe simple de séparation de données (Documents) et de leurs représentations (Vues).

Pour des raisons de performances, les documents et les vues s'exécutent dans le même contexte. Afin d'obtenir un design propre du logicielRx, les données et les algorithmes qui les utilisent ou les modifient sont rigoureusement séparés des méthodes d'affichage. Chaque action de l'utilisateur affecte premièrement la Vue. Cette dernière signale au Document les changements de données en conséquence. Puis, le Document doit informer toutes les Vue d'une mise-à-jour imminente de toutes les fenêtres afin de correctement refléter ces changements.

3.1.4 Automatisations Spécifiques à Rx

Le quatrième module est une implémentation spécifique au problème Rx Cervicales. Il s'agit de la large gamme de supports pour l'identification de contours, la segmentation d'images. Ce module est également chargé de générer des rapports de comparaisons d'angles. D'autres applications, notamment dans le domaine médical, sont parfaitement envisageables sur cette base.

fig. 3.3 - segment symétriques

3.2 Implémentation

3.2.1 Base Vectorielle

Un segment simple entre deux points (fig 3.1) est l'élément de base de Rx. Pour le dessiner, il suffit de sélectionner l'outil adéquat de la barre d'outils (fig 3.22) d'appuyer sur le bouton gauche de la souris et de glisser cette dernière à la position du point final. En relâchant, l'utilisateur obtient un segment de droite qu'il peut modifier en sélectionnant les extrémités et en les déplaçant avec un outil de modification de nœuds Il est également permis de déplacer tout le segment en le sélectionnant en dehors des coins et en le glissant à la nouvelle position choisie.

fig. 3.4a - angle (même segment)

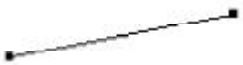

fig. 3.1 - segment simple

Des facilités de déplacement et de création ont également été implantées. En maintenant le bouton Shift, il est possible de créer un segment horizontal ou vertical ou encore de déplacer un point ou un élément vectoriel selon un seul axe.

Le principe utilisé pour un segment simple est facilement étendu à plusieurs points reliés. Le terme utilisé est de segments enchaînés (fig 3.2). Pour créer l'élément de la figure 3.2, il suffit de sélectionner l'outils de segments enchaînés et de placer un nombre illimité de points sur un document Rx. Le dernier point sera défini par un click du bouton droit de la souris. Les angles sont automatiquement calculés et affichés entre chaque segment de la suite enchaînée. (Une option permet bien évidemment de ne pas voir ces angles sur le document.)

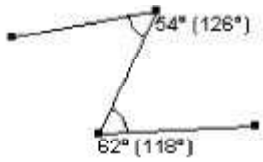

fig. 3.2 - segments enchaînés

Il est possible de joindre des points sur des segments. Il suffit de glisser un point sur un autre. Les sélections suivantes déplaceront les deux points ainsi mis en commun. Ceci permet de créer quasiment toute figure polygonale.

D'autres combinaisons de segments élémentaires sont disponibles. Il s'agit de segments à trois points symétriques autour du point central (fig 3.3) et d'un simple segment de trois points définissant un angle (fig.3.4a).

Les distances sont mesurées grâce à des segments de mesure (fig 3.5). Ces derniers peuvent être simples ou enchaînés (fig 3.6) tout comme les segments normaux.

Par défaut, un pixel équivaut à 1 mm. Ceci peut être facilement modifié avec un outil de Rx qui calcule automatiquement les rapports de tailles et ajuste les informations de tous les autres éléments (fig 3.7).

Le calcul d'angles se fait automatiquement pour toutes les intersections entre segments en utilisant le développement mathématique standard. Un exemple d'intersection entre deux segments est donne sur la figure 3.4b. Un autre aspect notable relève du calcul de la distance entre un point et un segment. Ceci est nécessaire pour la sélection d'éléments vectoriels par l'utilisateur. Les routines utilisées sont entièrement données dans l'Annexe A.

Les annotations textuelles sur une ligne (fig. 3.8) sont aussi des éléments vectoriels de Rx et réagissent a toutes les sélections, déplacements et suppressions. Il est important de pouvoir annoter le travail du médecin, ainsi Rx implémente le système d'un mini éditeur de texte sur une ligne. Il est possible de naviguer a l'intérieur de l'annotation et de l'éditer en tout moment. Les annotations peuvent avoir une police de caractère quelconque disponible dans le système ainsi qu'une taille variable. Tout ceci peut bien naturellement être modifie a tout moment. Les annotations sont transparentes et peuvent comme tous les autres éléments vectoriels changer de couleur pour plus de lisibilité.

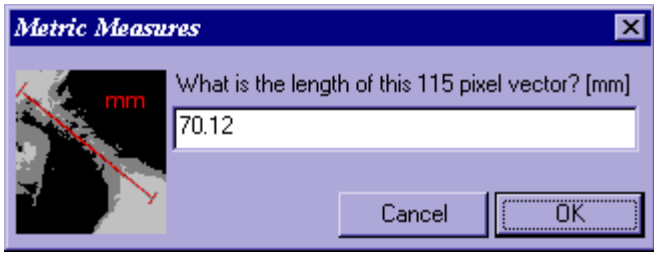

fig. 3.7 - définition du rapport métrique

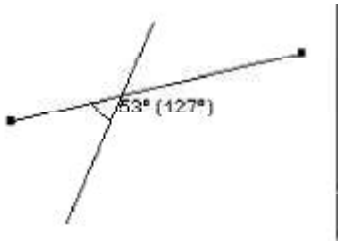

fig. 3.4b - angle
(intersection de deux segments)

En se basant sur le modèle des annotations, il est possible d'envisager des éléments plus complexes tels qu'un éditeur de format enrichi, de formules mathématiques ou de tableaux multidimensionnels.

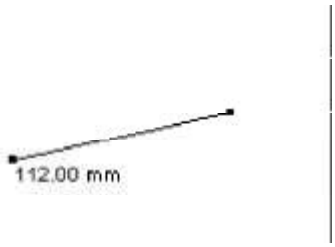

fig. 3.5 - segment de mesure

3.2.2 Support Bitmaps

Les images ou bitmaps (fig 3.9) dans Rx Cervicales sont des éléments semi-vectoriels. En effet, ces derniers réagissent a tous les traitements standards tels que les déplacements ou les rotations. Pour dimensionner une image il suffit de sélectionner un coin ou un point central d'une arête (un "anchor") et de le glisser dans n'importe quel sens. Les images sont transparentes. Il est ainsi possible de superposer plusieurs calques et de comparer les courbures en ajustant l'angle de rotation.

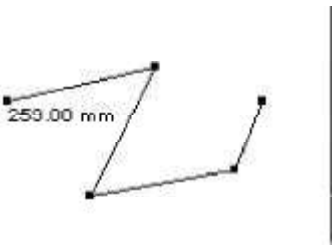

fig. 3.6 - segments de
mesure enchaînés

fig. 3.8 - annotation textuelle

Pour tourner une image, il faut déplacer un anchor en maintenant le bouton Ctrl. Il est également possible de tourner une image avec l'outil de rotation sélectionné sur la barre d'outils (fig 3.22) et maintenant la souris appuyée sur n'importe quel endroit de la surface de la page. Les figures

3.10 et 3.11 illustrent respectivement une image après élargissement ou réduction, puis après rotation.

fig. 3.9 - image (bitmap)

La rotation de bitmaps se fait par l'une des méthodes classique mais d'une manière inverse. En effet, il faut d'abord définir les dimensions de l'image après rotation, puis pour chaque point de l'image de destination calculer le point de l'image initiale en utilisant l'angle (alpha) de la rotation:

$$source_x = x \$\cos(a) + y \$\sin(a)$$

$$source_y = y \$\cos(a) - x \$\sin(a)$$

En pratique, la routine de rotation est plus compliquée Il faut tenir compte des transformations de coordonnées, des fenêtres d'affichage, mais surtout de la profondeur de l'image, car le calcu des masques de couleur différentes pour chaque profondeur de l'image est indispensable pour préserver une qualité raisonnable.

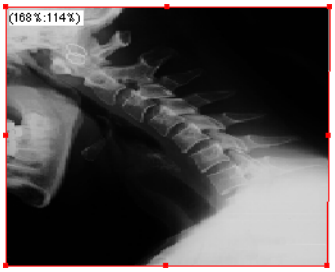

fig. 3.10 - élargissement d'une image

Les images que Rx Cervicales doit traiter sont importées à l'aide de commandes appropriées ou en utilisant le copier-colle depuis d'autres applications Windows. L'affichage tien rigoureusement compte des palettes de couleurs et le programme maintient les images en interne en tant que DIBs (Device Independent Bitmap) afin de garder une liberté envers le système de mesure utilise par les vues. De plus, une copie du buffer de sortie est également pre-calculée après chaque modification du contenu tel que la rotation, la réduction ou l'élargissement, qui sont très coûteux en ressources et temps de calcul.

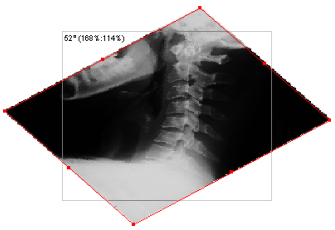

fig. 3.11 - réduction et rotation d'une image

Pour des raisons de performance Rx propose des options d'affichage de contenu de l'image pendant la rotation, la réduction ou l'élargissement. Toutefois, le médecin veut pouvoir ajuster une vertèbre sur l'autre dans les cliches de flexion et d'extension en faisant pivoter l'une des deux radios. Si l'image se dessine uniquement lorsque la rotation, la réduction ou l'élargissement sont termines cette opération devient très difficile. Par conséquent, Rx utilise un timer asynchrone qui traverse tous les managers qui doivent produire des images en sortie toutes les secondes et invalide les régions en rotation, en réduction ou en élargissement.

Une série de primitives sont développées sur les images DIB utilisées par Rx. Il s'agit d'implémentations d'algorithmes standards de conversions de profondeur, par exemple celle en niveaux de gris. Egalement, Rx propose des algorithmes tels que le Floyd Steinberg pour la conversion en Duotone. Il est aussi possible d'effectuer des manipulations simples telles que l'inversion de l'image.

3.2.3 Opérateurs Avancés sur les Bitmaps

Pour préparer une base solide aux automatisations angulaires (il faudra pouvoir reconnaître les contours des vertèbres sur les calques du médecin), des routines plus complexes sont présentes dans l'implémentation des DIBs de Rx. Tout d'abord, la base de

toute transformation est la convolution. La convolution discrète 2D avec une image originale *x(k,l)* et un filtre *g(k,l)* avec des dimensions respectives *N*, est définie par

$$y(k,l) = x(k,l) \&\&g(k,l) = \sum_{m=k-N+1}^{k} \sum_{n=l-N_g+1}^{l} x(m,n)g(k-m,l-n)$$

Elle permet d'isoler les contours et de faire ressortir des différences subtiles grâce a des matrice de convolution appropriées. La routine de convolution utilise une matrice de coefficients (opérateur de convolution) et est donnée dans sa version complète dans l'Annexe A.

En utilisant la convolution il est possible de spécifier des opérations telles que"Blur" ou "Sharpen Edges". Il serait également possible d'utiliser la convolution pour le filtre de Sobel ou Roberts avec des matrices appropriées. Toutefois, une version plus directe et rapide est écrite pour chacun de ces filtres en particulier.

Finalement, Rx propose la liste d'opérateurs sur des images suivante, certaines transformations sont illustrées dans les figures 3.12 a 3.17. Ces transformations sont:

- Elargissement ou réduction libre et proportionnelle a la taille originale de l'image (fig. 3.10); il faut noter que cette manipulation s'effectue toujours sur l'image originale, ainsi la qualité de l'image n'est pas diminuée lors de réductions et des élargissements consécutifs.
- Rotation libre autour du centre (fig. 3.11), détermine par croisement des diagonales du rectangle renfermant l'image en utilisant la formule de transformation inverse donnée ci-dessus.
- Miroir horizontal et vertical par simple inversion de bits del'image.
- Transformation en N/B, Bilevel ou Duotone (par seuillage) (fig. 3.12); la valeur moyenne des niveaux de gris est la somme des valeurs de l'histogramme divisée par le nombre total de points.
- Blur (fig. 3.13), convolution avec matrice $\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$.
- Blur more, convolution avec matrice $\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$.
- Transformation en N/B par diffusion, Floyd Steinberg (fig. 3.15); la requantification de l'image en diffusant l'erreur de la requantification sur les points voisins.
- Transformation en niveaux de gris (moyenne, linéaire) en calculant la moyenne entre les valeurs RGB avec des coefficients de $\frac{1}{3}$.
- Transformation en niveaux de gris (coefficients logarithmiques).
- Inversion (y compris images couleur, toutes les profondeurs) (fig. 3.16).

fig. 3.12 - conversion noir-blanc (bilevel)

fig. 3.13 - flou (blur)

fig. 3.14 - rehaussement de contours (sharpen edges)

- Sharpen edges (fig. 3.14), convolution avec matrice $\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$.

- Moyenne sur dimension variable, réduction de bruit (fig. 3.17); le principe du filtre est pour chaque pixel de trier les pixels voisins, puis de remplacer ce pixel par la valeur la plus proche de la moyenne des voisins triés.

- Amincissement de facteur constant; l'algorithme de thinning classique est utilise a une différence près: on détruit les pixels même si ces dernières constituent une ligne de bord fine.

- Opérateur de Roberts (Roberts Edge Detect) (fig. 3.20) utilise pour détecter le contour en tant que transition brusque de luminance ou de chrominance; pour une image x(k,l), l'opérateur de Roberts est donne par

$$h_{Roberts} = \left| \tilde{}_{Roberts(k,l)} \right| := \left| \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix} \right| + \left| \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix} \right|.$$

- Opérateur de Sobel (Sobel Edge Detect) (fig. 3.19) utilise pour détecter le contour en tant que extremum de la dérivée première; pour une image x(k,l), l'opérateur de Sobel est donne par

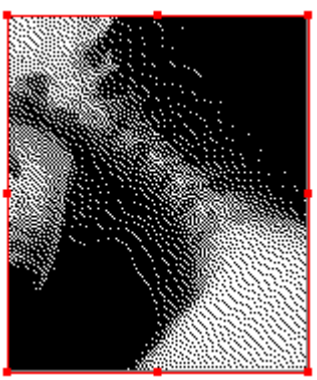

fig. 3.15 - conversion noir-blanc algorithme de Floyd-Steinberg

fig. 3.16 - inversion

fig. 3.17 - filtre de moyenne

fig. 3.18 - amincissement

fig. 3.19 - opérateur de Sobel

fig. 3.20 - opérateur de Roberts

$$h_{Sobel} = \left| \tilde{}_{Sobel(k,l)} \right| := \left| \begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix} \right| + \left| \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix} \right|.$$

- Opérateur laplacien, détectant un passage par zéro de la dérivée seconde; la dérivée seconde peut être approchée par des différents opérateurs laplaciens similaires

$$\Delta x(k,l) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \Delta x(k,l) = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad \Delta x(k,l) = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

Des informations sont disponibles sur des objets bitmaps. Rx affiche également des histogrammes pour l'image sélectionnée (fig. 3.21). Le premier histogramme représente la loi normale - la quantité de chaque niveau de gris dans l'image:$h(x_i)$.

Etant donnée le nombre total N_{tot} de points de l'image, la loi de densité de probabilité (probability density function - PDF) des niveaux de l'image, c'est a dire leur fréquence relative d'apparition est donnée par:

$$p_x(x_i) = \frac{h(x_i)}{N_{tot}}$$

qui vérifie bien

$$\Sigma_{niveaux_i} = p_x(x_i) = 1$$

La loi cumulative représentée dans le second histogramme est la somme des quantités de chaque niveau de gris inférieur normalisée, donnée par l'intégrale

Les dimensions de l'image affiches respectent également les métriques utilisées. Le trait rouge du premier histogramme est la valeur moyenne des niveaux de gris.

3.2.4 Interface Utilisateur

Comme pratiquement toute interface standard sous Windows, Rx utilise le système d'un menu contextuel, une barre d'état et une série de boutons pour la sélection des outils

disponibles. Des aides supplémentaires telles que des "ToolTips" ou encore la description de chaque commande dans la barre d'états sont disponibles en standard et permettent de simplifier la première approche d'un utilisateu novice. La barre d'outils (fig. 3.22) de Rx comporte tous les outils standard.

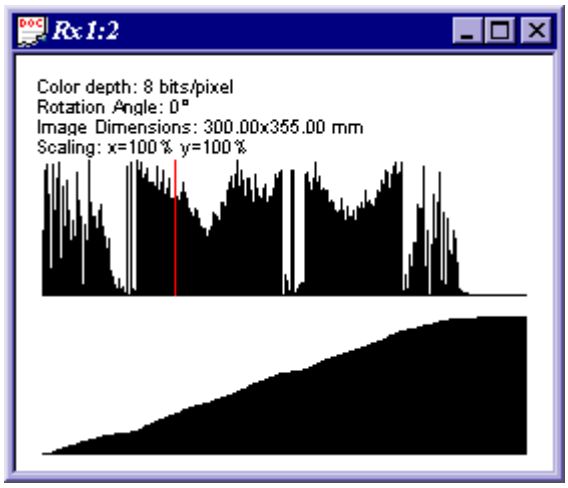

fig. 3..21 - histogramme de l'image de la fig. 3.9

L'architecture Document/Vue de Microsoft offre une approche intéressante a la mise-a-jour des éléments actifs ou des menus elle cache une majeure partie du système de messages tout en restant a un niveau très bas d'abstraction. En effet, un callback doit être crée pour chaque identificateur unique, obligatoire pour tout élément de l'interface Ceci permet d'assigner une valeur commune pour un menu et un bouton minimisant largement le code de l'interface.

Il est possible de sauvegarder, puis de charger des documents Rx. Il existe de nombreux formats de documents d'imagerie médicale tels que Papyrus très fréquemment utilise; car il permet d'associer des dossiers des patients aux images). Toutefois, il est nettement plus simple et rapide d'utiliser la serialisation, c'est la solution retenue. La serialisation consiste a stocker toutes les informations nécessaires a la création d'un objet quelconque et ceci pour toutes les données sans distinction. Par exemple, un segment de Rx nécessite la sauvegarde de la quantité de points qu'il possède et des valeurs x et y de chaque point individuellement. Les rares désavantages de cette méthode sont tout d'abord la modification du format Rx avec le rajout de nouveaux outils, puis de la taille rapidement croissante des documents. Un des objectifs d'un éventuel travail visant a améliorer ce logiciel serait l'utilisation d'un des formats tel que Papyrus.

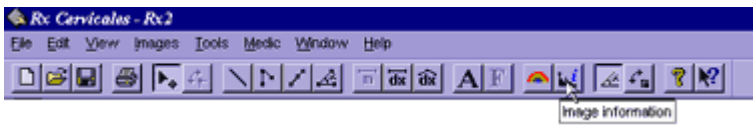

fig. 3.22 - barre d'outils de Rx

Le système d'affichage de Rx correspond dans les grandes lignes au modèle Document/Vue des MFC. Certains aspects sont toutefois intéressants a relever.

Le double buffering, a savoir l'affichage de tous les éléments sur une surface interne, puis la copie de cette dernière sur l'écran, est omniprésent. Ceci permet d'éviter le clignotement lors des rafraîchissements, de ne dessiner que des zones invalidées grâce a un système de clipping (il s'agit de ne redessiner qu'une région minimale) et ainsi d'accélérer considérablement le logiciel.

Ce système est implante dans la base de toute classe d'affichage de Rx comme la fenêtre principale et les histogrammes. Ceci rend le travail du médecin plus agréable par des opérations sur des objets affiches sans clignotement.

Il existe sous Windows pas moins de huit métriques différentes Ces dernières sont pratiques dans certains cas, mais amènent le plus souvent des ambiguïtés considérables pour le programmeur compte tenu de l'absence complète de coordination entre des mesures de pages d'imprimante, de résolutions d'imprimantes et de l'affichage graphique. Certains principes

relèvent d'un illogisme, par exemple la définition des tailles de documents et des vues sur ces derniers. En occurrence la taille du document retournée par une fenêtre de mise-en-page dépend de la résolution de l'imprimante. En revanche il est impossible de savoir quelle est cette dernière. Pire, la scalabilite des fontes n'est pas linéairement dépendante de tous ces paramètres rendant difficile voire impossible le placement d'un texte par rapport a des bitmaps. Il est ainsi extrêmement difficile de développer un réel logicielWYSIWIG (“What You See Is What You Get”) sans recréer de nombreuses couches d'abstraction a ce système natif et de créer un système d'affichage et d'impression propriétaire

Malgré tous ces points négatifs du système, certains problèmes ont étés évites d'une manière très directe. L'impression, par exemple, n'est que celle du double buffer de l'affichage complet redimentionne sur le device context (l'élément de base de toute surface d'affichage) du travai d'impression et la taille des documents utilise des facteurs de réduction ou d'élargissement recalculés

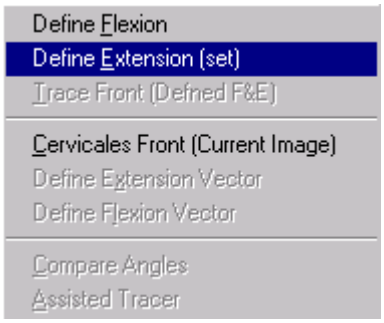

fig. 3..23 -menu Automatisations Angulaires

3.2.5 Automatisations Angulaires

Sur la base d'un développement d'éléments vectoriels de bitmaps, des opérations sur ces derniers telles que des rotations et des transformations planes il est possible de construire des outils puissants et varies dans le domaine d'imagerie médicale Ce travail offre un exemple parmi bien d'autres, i.e. une application a la mesure sur des radiographies cervicales fonctionnelles. Dans un travail futur, on peut imaginer une interface de programmation permettant de développer des "plugins" a l'image des automatisations angulaires de ce projet.

Le problème pose par ce travail nécessite plusieurs étapes:

- définir les deux images de flexion et d'extension
- nettoyer ces images, les convertir en bilevel (deux couleurs)
- identifier des zones de surface importantes (vertèbres)
- segmenter ces zones, générer des segments droits et ne garder que sept ensembles de segments (sept vertèbres) de longueur maximale
- identifier le mur vertébral frontal
- calculer et comparer les angles entre les cliches de flexion et d'extension

Définitions des Images

Rx permet de travailler sur un grand nombre de bitmaps. Le médecin peut définir les images correspondantes au calque de flexion et d'extension a l'aide du menu correspondant (fig. 3.23). Il peut évidemment redéfinir ces choix et ne peut définir le même calque pour les deux catégories.

Filtres et Transformations

Les calques numérisés par le médecin présentent une quantité de bruitnon négligeable. Il s'agit d'un bruit non-uniforme et souvent de contraste très faible, i.e. dans les teintes des niveaux de gris très inférieurs a une valeur moyenne, utilisée pour leseuillage, souvent très proche du noir.

Pour se débarrasser du bruit plus ponctuel, i.e. des zones de gris visibles a l'œil nu on applique aux deux calques un filtrer de moyenne (fig 3.17). Puis, on utilise un simple seuillage pour transformer cette image en deux couleurs. La figure 3.24 montre une partie du calque original, puis ce dernier seuillé sans filtre de moyenne et enfin seuille après le filtre de moyenne. On remarque la différence du bruit résultant notable.

Des essais ont étés effectues en utilisant le filtre de Roberts (fig. 3.25). Ce dernier donne en théorie et en pratique d'excellents résultats Mais il détecte un contour fin autour des zones noires et il est très difficile d'exploiter ce résultat

Identification de Vertèbres

En parcourant toute l'image et pour chaque point noir, Rx cherche toute la surface qui contient ce dernier. La poursuite des zones noires permet de trouver les vertèbres individuellement et de transformer des surfaces en des listes de points. Au maximum 12 zones noires sont retenues (cette valeur a été déterminée statistiquement) éliminant une partie importante d'erreurs dues aux annotations éventuelles sur le calque ou encore au bruit induit par le scanner laisse après le passage des filtres.

Segmentation et Fragmentation

Pour chaque zone noire Rx applique l'algorithme standard de fragmentation linéaire récursive afin de simplifier des zones définies par les étapes précédentes en des vecteurs de idéalement 4 points. En sélectionnant deux points, Rx trace un segment entre eux et recherche le point le plus éloigné de ce dernier (erreur maximale) dans la liste des points restants. Puis, par itération on recherche tous les points tant que la distance dépasse 0.25% de la largeur de l'image. Ceci permet d'éliminer les points sur la même droite et constitue une première étape de vectorisation. Le principe est illustre sur la figure 3.26.

Puis, seules 7 des 12 zones ayant une surface intérieure maximale sont gardées Elles constituent les 7 vertèbres qui présentent un intérêt

Identification du Mur Vertébral

L'identification du mur frontal vertébral peut se faire après que les points les plus proches sont réduits a leur moyenne (puisque Rx s'intéresse au mur intérieur, le point le plus a gauche sera garde). Les deux points ayant la valeur la plus petite selon l'axe des x constituent le mur frontal de la vertèbre en traitement. De plus, Rx trie les points selon la verticale pour garantir un minimum d'écart du a une erreur éventuelle ou a un calque incomplet.

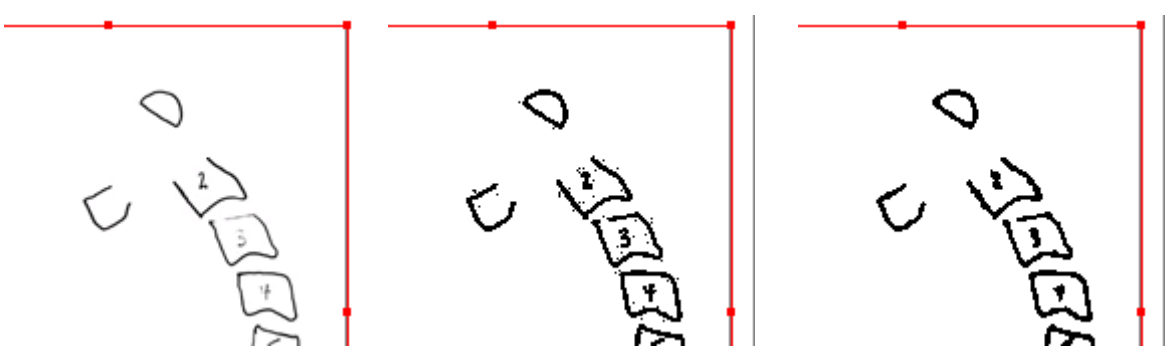

fig. 3.24 - calques original, seuillé, seuillé après moyenne

Toutefois, des erreurs peuvent survenir lors de cette identification. Le médecin peut toujours les corriger en déplaçant chaque point individuellement. La qualité des résultats est discutée dans le chapitre suivant.

Calculs et Comparaisons d'Angles

Les mesures automatique d'angles sont faites a partir d'un vecteur passant par les 14 points du mur cervical frontal détecte a l'étape précédente (fig. 3.27). Rx affiche les angles entre les calques en flexion et en extension ainsi que les distances entre les vertèbres consécutives tout en tenant compte des métriques définies par l'utilisateur. Il est possible de déplacer des points de ce mur et de recalculer les angles corrigeant ainsi les erreurs du trace de contours.

Le médecin peut exporter ces données angulaires dans le presse-papier de Windows et ainsi les utiliser pour des comparaisons avec des sujets sains. Une impression est a prévoir dans une version future.

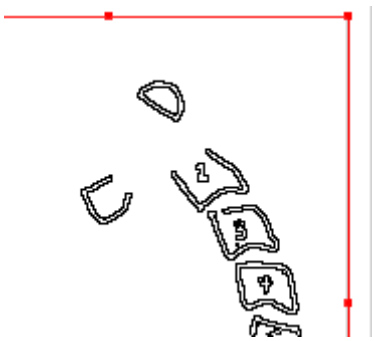

fig. 3..25 - opérateur de Roberts sur calque

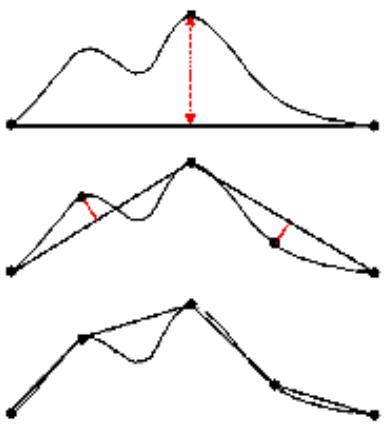

fig. 3.26 - fragmentation

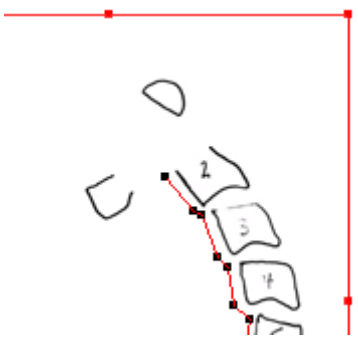

fig. 3..27 - mûr cervical frontal

Chapitre 4

4 Résultats et Conclusion

4.1 Résultats

4.1.1 Contexte

Rx-Cervicales a été testé dans le cadre de la Clinique de Rééducation de l'Hôpital Cantonal de Genève par Dr. Paul Cohen pendant une période de trois semaines. 12 calques produits par ce dernier à partir de radiographies de cervicales ont étés analysés par le programme. Nous présentons ci-dessous à l'aide d'un exemple concret d'une patiente blessée dans un accident de voiture les étapes que le médecin a suivi afin de tester le logiciel. Le choix de l'échantillon et les résultats complets sur sont présentés plus loin.

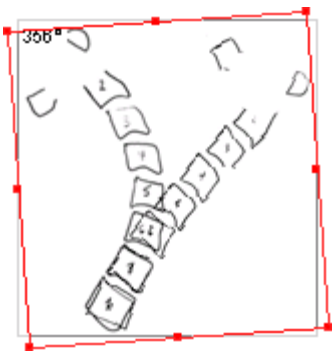

fig. 4.2 - images superposés

Alors qu'elle s'arrête à un feu rouge, la patiente de 35 ans subit un choc à l'arrière de son véhicule. Son appui-tête étant mal réglé, sa tête part en arrière dans un mouvement d'hyperextension. Il n'y a pas de choc direct sur le crâne et la patiente ne perd pas connaissance. Elle décrit au médecins l'apparition rapide de douleurs de la colonne cervicale qui l'empêchent de tourner normalement la tête. La patiente se rend dans un service d'urgence où un bilan clinique et radiologique ne montre aucune lésion vertébrale ni aucun déficit neurologique. La patiente est autorisée à rentrer à domicile sous protection d'une minerve. Dans les mois qui suivent, les douleurs cervicales diminuent mais sans disparaître. Un bilan radiologique cervical fonctionnel réalisé après plusieurs mois mettra en évidence une instabilité entre C3 et C4 (troisième et quatrième vertèbres cervicales). En raison de la persistance de douleurs cervicales accompagnées de fréquents maux de tête, un nouveau bilan cervical radiologique fonctionnel sera effectué après quelques mois et montrera la persistance de l'instabilité. La patiente sera finalement opérée au niveau instable.

fig. 4.1 - radiographies en flexion et en extension

La figure 4.1 montre les radiographies originales de cette patiente en flexion et en extension.

4.1.2 Clichés et Calques

Puisque le matériel que Dr. Paul Cohen ne permet pas de numériser directement une radiographie (ceci nécessiterait un scanner à transparents), il produit un calque pour la radiographie de la colonne cervicale en flexion et en extension. Après avoir utilisé un logiciel fourni avec le scanner pour convertir ces images en un

format reconnu par Rx-Cervicales, le médecin les importe dans ce dernier à l'aide de commandes appropriées.

Rx-Cervicales est capable d'afficher les deux calques en transparence, de les superposer et les faire pivoter afin d'ajuster la première vertèbre (fig. 4.2). Cette étape est déjà une avancée considérable pour le médecin, car il peut détecter à l'œil nu des différences subtiles entre les deux positions du cou qui ne sont pas visibles sur la radio d'origine. De plus, les outils de Rx permettent de visualiser un nombre bien plus important de détails de tracer rapidement des droites et en mesurer des écarts ou des angles.

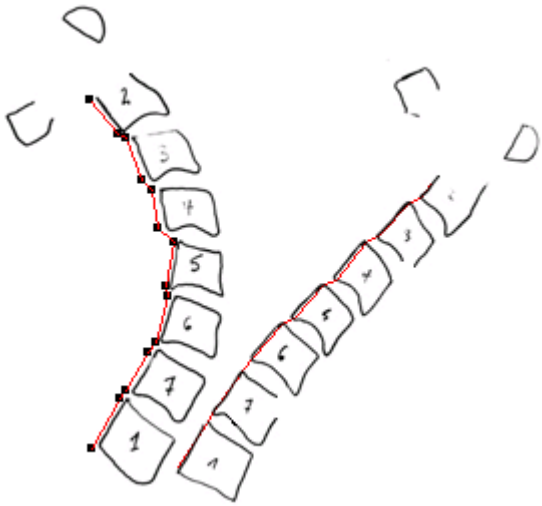

fig. 4.3 - mûr frontal cervical produit par Rx-Cervicales

4.1.3 Automatisations Angulaires

Après avoir importé les deux calques, le médecin définit le cliché de flexion et celui d'extension. Rx est désormais prêt à tracer les vecteurs définissant les murs intérieurs des deux images. Rx produit ainsi deux vecteurs de 14 points, correspondants bien aux extrémités des vertèbres (fig. 4.3).

Un rapport est automatiquement affiché pour les écarts angulaires et les distances entre les vertèbres (fig. 4.4). On remarque une valeur bien trop importante au niveau de C3, ce qui est bien le diagnostic du médecin.

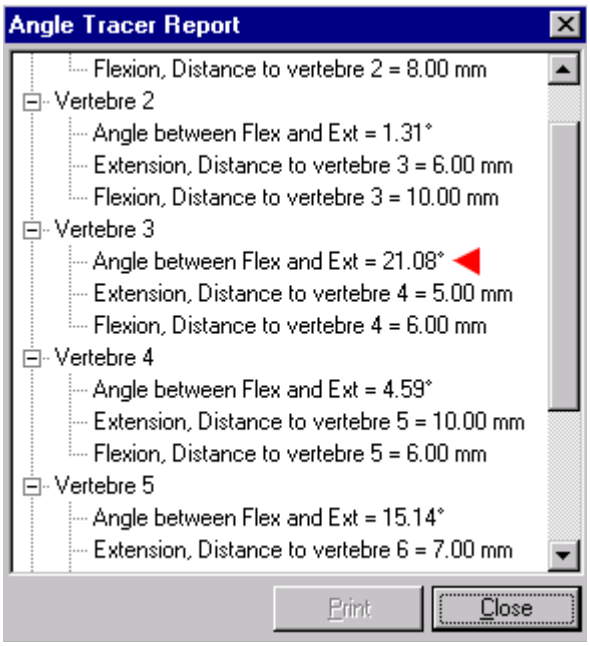

fig. 4.4 - rapport angulaire

4.1.4 Résultats

Pour tester la qualité des résultats de Rx sur l'échantillon de patients décrit dans 4.1.1, la majorité des calques produits par Dr. Paul Cohen ont étés tracés avec un qualité inférieure à celle qui sera normalement utilisée. Ainsi, les résultats présentés ci-dessous sont les plus pessimistes.

Par calque de mauvaise qualité on entend des images d'une petite taille avec beaucoup de bruit non-uniforme et ajouté artificiellement. De plus, des annotations manuelles annotant chaque vertèbre constituent un excellent test de l'algorithme de fragmentation de Rx, i.e. le choix de zones d'influence pour la reconnaissance du mûr frontal cervical.

Sur l'échantillon choisi, seuls deux calques n'ont pas produit un vecteur du mûr frontal correct. Dans ces deux cas, le premier possédait un point du mûr frontal décalé considérablement et le second seulement de quelques

pixels (fig. 4.5). Ces erreurs proviennent principalement du fait que le calques possèdent des zones noires non-continues. En outre, des vertèbres collées l'une à l'autre d'une manière significative empêchent l programme de les considérer comme deux surface séparées. Toutefois, 85% des calques donnent le résultat souhaité.

D'autres erreurs sont prévisibles, notamment il n'est pas possible de savoir dans quelle direction une vertèbre est tournée sur le cliché d'extension. Une convention à été adoptée avec le médecin afin de bien choisir le coin du mûr frontal lors de litiges.

Les angles mesurées correspondent bien à la réalité observée et le diagnostic du médecin. Pour le calcul d'angles l'ordinateur doit faire des arrondis au pixel près mais les mesures faites à la main génèren souvent de bien pires erreurs. Il faut toutefois noter que les chirurgien font toutes les mesures d'une manière beaucoup plus minutieuse avan toute intervention et Rx sert à donner une indication au médecin lors d la première observation du patient.

fig. 4.5 - faux résultat

De plus d'excellents pourcentages de succès au niveau de la reconnaissance du mûr frontal cervical, certains aspects de Rx ont étés agréablement surprenants. Par exemple, la vitesse des filtres même sur des machines lentes dépasse largement les attentes. Il s'agit de toutes les transformations sur les bitmaps et également des détections de contours.

4.2 Evolutions Futures

Le logiciel Rx peut être considéré comme un premier pas vers un système beaucoup plus évolué traitant des images médicales ainsi que des éléments vectoriels - deux concepts indissociables dans un travail de médecin mais largement omis dans les programmes du même genre. Les radiographies sont sous-exploitées et une partie importante de temps est utilisée pour une analyse systématique pouvant être faite par un ordinateur avec des logiciels appropriés, voire par des machines dédiées.

Cette conclusion ouvre de nombreuses portes à l'évolution de Rx. Tout d'abord il sera important de pouvoir traiter directement les images digitales produites par les scanners de l'hôpital. Ceci passe par la possibilité de segmenter des images de radios et non plus des calques. Rx devrait pouvoir importer et d'exporter une gamme variée d'images de formats existants.

Une possibilité de scanner directement à partir de Rx serait un plus, une modification d'images et une palette d'outils d'un logiciel bitmap standard pourra être utile. Un système d'annulation d'opérations précédentes (undo) semble indispensable.

Une interface API permettrait d'écrire des plugins aisément poussant le développement d'outils de plus en plus complexes. Un langage macro ou script ajouterait une étape intermédiaire et un personnel peu technique pourrait automatiser des tâches systématiques.

4.3 Conclusion

Tout d'abord, les objectifs qui ont étés fixés ont étés largement atteints. Rx-Cervicales permet de travailler sur des éléments vectoriels tels que des droites simples et des segments joints ainsi que sur des objets semi-vectoriels tels que des bitmaps. L'utilisateur peut définir des échelles de mesure pour des longueurs de segments, de pivoter et de redimensionner les images. La vaste gamme de filtres proposée sur les bitmaps offre de nombreuses possibilités et constitue une base solide pour les automatisations et les calculs sur les radiographies. Finalement, l'automatisation des comparaisons sur les calques de flexion et d'extension de la colonne cervicale ont étés largement atteints. De plus, Rx dépasse certaines espérances au niveau de la qualité de reconnaissance ou encore de la vitesse des filtres.

Un autre aspect important est celui de la disponibilité du code source[7] et du champ couvert par le programme, notamment au niveau du traitement d'images. Certaines routines ont nécessité un travail considérable et leur utilisation faciliterai des développements futurs similaires.

5 Références, Bibliographie, Annexes

5.1 Références, Bibliographie

5.1.1 Imagerie Numérique

- Théo Pavidis, Algorithms for Graphics and Image Processing, Computer Science Press, 1982 [PAV]
- Thierry Pun, Imagerie Numérique, Cours, Groupe de Vision par Ordinateur, Département d'Informatique, Université de Genève, Suisse, 1996[PUN]
- Scott E. Umbauch, Computer Vision Image Processing, A Practical Approach using CVIPtools, Prentice Hall PTR, 1998 [CVIP]

5.1.2 Programmation

- David Kruglinsky, Inside Visual C++, Microsoft Press, 1997 [KR]
- MasterClass Visual C++, Wrox Press, 1995 [WROX]
- Scott Stanfield, Visual C++ How-To, Waite Group Press, 1995 [HOW]

5.1.3 Autres

- Michael P. D'Allessandro, M.D. http://www.medicalstudent.com [STUD]
- Virtual Hospital, http://www.vh.org, University of Iowa, College of Medecine, Department of Radiology, USA [VH]

5.2 Remerciements

- Prof. Thierry Pun, Université de Genève, groupe Vision par Ordinateur
- Dr. Paul Cohen, Clinique de Rééducation de l'Hôpital Cantonal de Genève
- Prof. Juergen Harms, Université de Genève
- Pierre Sabatzus, Dataweb S.A, Genève
- Jean-Luc Wichoud, Université de Genève, faculté de Mathématiques
- Lori Petrucci, Université de Genève, groupe Vision par Ordinateur
- et à toutes les autres personnes qui ont apporté leur soutien ou leur savoir tout au long de l'évolution de ce projet

Annexe A - Code Cité

```
/*
  Intersection entre deux droites données par
  (x1, x2) (y1, y2) et (z1, z2) (w1, w2)
```

```
  */

int CAVector::Intersection(double x1,double x2,
                           double y1,double y2,
                           double z1,double z2,
                           double w1, double w2,
                           CPoint& Target){
  double r,v1,v2,k,l,vtn1,vtn2;
  v1=y1-x1;
  v2=y2-x2;
  if (!v1) v1=0.00000000001;
  if (!v2) v2=0.00000000001;
  r=sqrt((v1*v1)+(v2*v2));
  vtn1=w1-z1;
  vtn2=w2-z2;
  if (!vtn1) vtn1=0.00000000001;
  if (!vtn2) vtn2=0.00000000001;
  k=(x1+((z2*v1)/v2)-((x2*v1)/v2)-z1)/
    (vtn1-((vtn2*v1)/v2));
  l=(z2+((vtn2*x1)/vtn1)-((z1*vtn2)/vtn1)-x2)/
    (v2-((vtn2*v1)/vtn1));
  if(l < 0.0) return 0;
  else if(l > 1.0) return 0;
  if(k < 0.0) return 0;
  else if(k > 1.0) return 0;
  Target.x = (long) (x1+l*v1);
  Target.y = (long) (x2+l*v2);
  return 1;
}

/*
   Angle en degrés entre deux droites données par
   (x1, x2) (y1, y2) et (y1, y2) (z1, z2)
   (y1, y2) est l'intersection
   */

double CAVector::AngleXYZ(double x1,double x2,
                          double y1,double y2,
                          double z1,double z2) {
 double r,v1,v2,k,l,vtn1,vtn2,
        i1,i2,a,b,res1,res2, res3;
 v1=y1-x1;
 v2=y2-x2;
 if (!v1) v1=DBL_EPSILON;
 if (!v2) v2=DBL_EPSILON;
 r=sqrt((v1*v1)+(v2*v2));
 if (v2 == DBL_EPSILON) vtn1=-DBL_EPSILON;
 else vtn1=-v2/r;
 if (v1 == DBL_EPSILON) vtn2=DBL_EPSILON;
 else vtn2=v1/r;
 k=(x1+((z2*v1)/v2)-((x2*v1)/v2)-z1)/
   (vtn1-((vtn2*v1)/v2));
 l=(z2+((vtn2*x1)/vtn1)-((z1*vtn2)/vtn1)-x2)/
   (v2-((vtn2*v1)/vtn1));
 i1=x1+l*v1;
 i2=x2+l*v2;
 a=sqrt(((x1-i1)*(x1-i1))+((x2-i2)*(x2-i2)));
 b=sqrt(((z1-i1)*(z1-i1))+((z2-i2)*(z2-i2)));
 res1=fabs(atan(b/a));
 if(l <= 0) res2=(M_PI-res1); else res2=res1;
 res3 = (res2/M_PI)*180;
 if (k >= 0) return -res3;
 else return res3;
}
```

```
/*
  Distance entre un segment donné par
  (x1,x2) (y1, y2) et un point (a, b)
  */

double CAVector::Distance(double x1, double x2,
                          double y1, double y2,
                          double a, double b) const {
  double v1,v2,vtn1,vtn2,k,l,r;
  if (y2 == x2) y2+=0.0001;
  v1 = y1-x1;
  v2 = y2-x2;
  r = sqrt((v1*v1)+(v2*v2));
  vtn1 = -v2/r;
  vtn2 = v1/r;
  k = (x1+((b*v1)/v2)-((x2*v1)/v2)-a)/
      (vtn1-((vtn2*v1)/v2));
  l = (b+((vtn2*x1)/vtn1)-((a*vtn2)/vtn1)-x2)/
      (v2-((vtn2*v1)/vtn1));
  if (l <= 0)
      return sqrt((x1-a)*(x1-a)+(x2-b)*(x2-b));
  else if (l >= 1.0)
      return sqrt((y1-a)*(y1-a)+(y2-b)*(y2-b));
  else return fabs(k);
}

/*
  Convolution sur un Bitmap
  */
void CDib::Convolve(float * i_Kernel,
        unsigned int i_MWidth, unsigned int i_MHeight){
/* image must be at least the size of the kernel */
   if ((unsigned int)
        m_lpBMIH->biWidth < i_MWidth) return;
   if ((unsigned int)
        m_lpBMIH->biHeight < i_MHeight) return;
   /* show progress dialog */
   theApp.ShowProgress("Performing convolve,
        please wait.");
   /* allocate the output image buffer */
   int l_newSize =
      m_lpBMIH->biHeight * (((
         (m_lpBMIH->biWidth * m_lpBMIH->biBitCount)
      + 31) & ~31) / 8);
   LPBYTE new_Image = new BYTE[l_newSize];
   /* fill missing pixels (convolve cuts) */
   memcpy(new_Image, m_lpImage, l_newSize);
   unsigned int l_ColOffset = i_MWidth / 2;
   unsigned int l_RowOffset = i_MHeight / 2;
   /* compensate edge defects */
   unsigned int l_Col = l_ColOffset;
   unsigned int l_Row = l_RowOffset;
   unsigned int l_Width =
        m_lpBMIH->biWidth - i_MWidth + 1;
   unsigned int l_Height =
        m_lpBMIH->biHeight - i_MHeight + 1;
   /* row range of pixels */
   unsigned int l_colExtent = l_Col + l_Width;
   unsigned int l_rowExtent = l_Row + l_Height;
   unsigned int l_TempRow, l_TempCol;
   double l_SumR, l_SumG, l_SumB;
   float * l_KernelPtr;
   unsigned int l_KernCol, l_KernRow;
```

```
   unsigned int l_ImageRow, l_ImageCol;
   DWORD l_Pixel;
   int nRowBytes =
        ((((m_lpBMIH->biWidth * m_lpBMIH->biBitCount)
      + 31) & ~31) / 8);
   float l_KernelByte;
   for (l_ImageRow = l_Row; l_ImageRow < l_rowExtent;
        l_ImageRow++) {
    l_TempRow = l_ImageRow - l_RowOffset;
    theApp.SetProgress(l_ImageRow * 100 / l_rowExtent);
    for (l_ImageCol = l_Col; l_ImageCol < l_colExtent;
        l_ImageCol++) {
      l_TempCol = l_ImageCol - l_ColOffset;
      l_SumR = 0;
      l_SumG = 0;
      l_SumB = 0;
      l_KernelPtr = i_Kernel;
      for (l_KernCol = 0; l_KernCol < i_MWidth;
        l_KernCol++)
      for (l_KernRow = 0; l_KernRow < i_MHeight;
        l_KernRow++) {

        l_Pixel = GetPixel(l_TempCol + l_KernCol,
                l_TempRow + l_KernRow,
                m_lpBMIH,
                m_lpImage,
                nRowBytes);
        l_KernelByte = *l_KernelPtr++;
        l_SumR += GetRValue(l_Pixel)*l_KernelByte;
        l_SumG += GetGValue(l_Pixel)*l_KernelByte;
        l_SumB += GetBValue(l_Pixel)*l_KernelByte;
        l_SumR = (l_SumR<0)?0:
                 (l_SumR>0xFF)?0xFF:l_SumR;
        l_SumG = (l_SumG<0)?0:
                 (l_SumG>0xFF)?0xFF:l_SumG;
        l_SumB = (l_SumB<0)?0:
                 (l_SumB>0xFF)?0xFF:l_SumB;
        }
       PutPixel(l_ImageCol, l_ImageRow,
        RGB(l_SumR, l_SumG, l_SumB),
        m_lpBMIH, new_Image, nRowBytes);
   }}
   delete m_lpImage;
   m_lpImage = new_Image;
   theApp.HideProgress();
   m_Invalid = TRUE;
}
```


[1] Statistiques du Touring Club Suisse
[2] Informations fournies par Dr. Paul Cohen, HCUG
[3] Centre d'Imagerie Médicale, Unité d'Imagerie Numérique
[4] Navimetric Tonometric 2D - http://www.navimetric.com
[5] Osiris - http://expasy.hcuge.ch/www/UIN/html1/projects/osiris/osiris.html
[6] PACS - http://expasy.hcuge.ch/www/UIN/html1/projects/pacs/pacs.html
[7] Les sources, les exécutables, ainsi que la version électronique de ce rapport sont disponibles sur http://cuiwww.unige.ch/~doubrov5/uni/licence