Research of New Medical Volume Visualization Methods: Application in the Kidney Preoperative Planning System

Hui Tang

N° ordre de la thèse : 3861

THÈSE

Présentée devant

L'UNIVERSITÉ DE RENNES 1

Pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : TRAITEMENT DU SIGNAL ET TÉLÉCOMMUNICATIONS

Par

TANG Hui

Équipes d'accueil :	LTSI, UMR INSERM U642, Université de Rennes 1, France LIST, SouthEast University, Nanjing, Chine CRIBS, Laboratoire International Associé
École Doctorale :	Mathématiques, Informatique, Signal, Électronique et Télécommunications
Composante universitaire :	UFR Structure et propriétés de la Matière

Méthodologies de visualisation 3D en imagerie médicale

Soutenue le 13 Décembre 2008 devant la commission d'examen composée par :

Président	Senhadji Lotfi	Pr, Université de Rennes 1, France
Rapporteurs :	Chen Wufan	Pr, Southern Medical University, Guangzhou, Chine
	Demongeot Jacques	Pr, Université Joseph Fourier, Grenoble, France
Directeurs :	Dillenseger Jean-Louis	MC HDR, Université de Rennes 1, France
	Luo Limin	Pr, SouthEast University, Nanjing, Chine
Examinateurs :	Shu Huazhong	Pr, SouthEast University, Nanjing, Chine
	Yang Jingyu	Pr, Nanjing University of Science and Technology, Nanjing, Chine
	Zhou Mingquan	Pr, Normal University, Beijing, Chine

Acknowledgments

This dissertation was performed in the frame of CRIBs, an international French-Chinese lab, Université Rennes I, France – INSERM, France – SouthEast University, Nanjing, China.

First and foremost, I would like to thank my supervisors, Prof. LUO Limin and Mr. DILLENSEGER Jean-Louis, who suggested this project and gave me encouragement and guidance throughout the investigation and preparation of this dissertation. Without their continuing inspiration, suggestions, criticism and support, this work would not have been possible. I also want to thank Mr. DILLENSEGER for his patient care about me when I was in France. The kindness of him and his family let me feel homey and removed my scare about a strange environment.

I would like to thank Prof. SHU Huazhong for the contact between the two labs and the proposition of some ideas for this dissertation. I would also like to thank Prof. BAO Xudong for his advice and help during my studies. When I was depressed, he encouraged me and gave me useful suggestions which helped me a lot. Thank you very much. I also appreciate all the help I received from the other professors and teachers in LIST: YU Wenxue, WANG Shijie, YANG Wu, ZHOU Weiping, LI Songyi, ZHANG Pinzheng, WANG Zhen et al.

When I was in LIST, I spent much time with my colleagues and friends: HU Yining, RONG Chengcheng, XIE Lizhe, YANG Guanyu, ZHANG Qiukui et al. Thank you for all good time we spent together. I spent the other half time of my PH.D studies in the French lab LTSI. I would also like to thank all people in this lab for the twenty months' good time. In particular, a big hug and gratitude to GARCIA Marie-Paule, ESNEAULT Simon, BOUSSE Alexandre, TSE Kevin, GARNIER Carole and JEHENNE Guillaume et al. for the happy time we spent together, for their endless help, for being friendly throughout the whole of my stay in France.

I cannot forget to thank my parents and my brother. They always support me and give me much courage to continue.

I reserve my highest appreciation for my husband YANG Kui for the love, for the care, for the support. He is the source of my strength. Whenever I have a problem, I know that he is always there.

This dissertation is dedicated to everyone who gave me moral, technical and material support for my PH.D studies.

TANG Hui

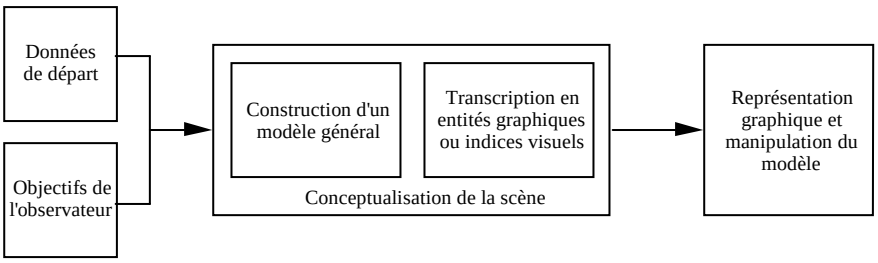
September, 2008

Résumé étendu du mémoire de Thèse

Chapitre 1 : Introduction

La visualisation est encore un champ de recherche relativement actif. Ceci vient du fait que la vision est le sens le plus développé chez l'homme. La vision et les capacités psychovisuelles et cognitives du cerveau qui en découlent, représentent pour l'homme l'outil d'analyse privilégié de son environnement. L'idée de base des techniques de visualisation consiste à exploiter les propriétés inhérentes du système visuel humain pour explorer, comprendre et analyser tout type de données, d'information ou de savoir produit par le monde scientifique Le domaine de recherche de la visualisation peut être défini par l'ensemble des techniques qui permettent de transcoder ou modéliser des phénomènes ou structures complexes, à grand nombre de variables (numériques à symboliques) et de dimensions élevées afin de pouvoir les manipuler et les analyser sous une forme visuelle, c'est-à-dire permettre à l'observateur de construire un modèle mental des processus d'une scène complexe.

Une étude précédente à permis de définir les axes majeurs permettant d'élaborer un outil de visualisation répondant à un besoin précis émis par un utilisateur [1]. Cette réflexion a essayé de faire ressortir les points principaux qui sont à prendre en compte lors de la conception d'un outil de visualisation ainsi que les domaines de compétences nécessaires aux développeurs.



Ainsi, trois points principaux sont à prendre en compte :

- La connaissance de la nature des données (dimension, nature et topologie spatiale des données) et surtout des objectifs de l'utilisateur (intérêt de certaines variables, corrélation entre données).
- La transposition ou conceptualisation sous une forme schématique et visuelle de l'information contenue dans les données et devant être représentée. Cette étape peut être divisée en deux sous-parties : 1) la construction d'un modèle général à partir des données et pouvant servir de représentation des phénomènes et 2) la transcription de ce modèle en entités graphiques ou indices visuels.
- La représentation graphique et la manipulation des éléments graphiques.

Du fait de sa complexité, le domaine médical, fait un grand usage d'outils de représentation. Toutefois, du fait de l'évolution rapide des techniques d'imagerie et de la miniaturisation des outils d'intervention, de nouveaux besoins se sont crées en terme de visualisation. Ces besoins recouvrent tant le domaine de l'imagerie de diagnostic (imagerie 3D +t) que l'imagerie interventionnelle avec des notions de planning opératoire, de guidage du geste et de fusion d'information dans le champ opératoire.

L'objectif de cette Thèse est de se focaliser sur certains des problèmes non résolus de la chaîne de visualisation en essayant d'y apporter quelques solutions. Toutefois, et cela est particulièrement vrai dans le domaine de recherche de la visualisation, l'élaboration de nouvelles solutions est largement liée au contexte de l'application finale et des données qui y sont traitées. Nous nous sommes intéressés à une application médicale bien précise qui est la chirurgie conservatrice des tumeurs rénales. Plus particulièrement, nos travaux de Thèse ont concerné l'élaboration des différents éléments d'un outil de visualisation pour le planning préopératoire de cette chirurgie rénale.

Pour cela, nous sommes partis du cadre général décrit précédemment permettant d'élaborer un outil de visualisation et l'avons adapté à la problématique médicale. Nous y avons identifié plusieurs points durs auxquels nous avons essayé d'y apporter des éléments de réponses. Ainsi :

- La connaissance de la nature des données et surtout des objectifs de l'utilisateur. L'examen préopératoire à la chirurgie des tumeurs rénales est l'uroscan. Il consiste à acquérir, par scanner X, 3 ou 4 volumes de l'abdomen à différents stades de diffusion de produit de contraste. Une première acquisition avant injection permet d'imager le volume du patient. Une seconde acquisition est prise juste après injection, ce qui révèle la vascularisation rénale artérielle et permet de discriminer la tumeur. Une troisième acquisition est effectuée juste après, elle présente un rehaussement du parenchyme rénal et permet également de discriminer la tumeur. Une dernière acquisition, effectuée 10 à 20 minutes après l'injection de produit de contraste, permet de visualiser le système collecteur urinaire. L'objectif est de confronter ces acquisitions afin de fournir au chirurgien le cadre anatomique spécifique du patient.

- La transposition ou conceptualisation sous une forme schématique et visuelle de l'information contenue dans les données et devant être représentée. Plusieurs étapes sont nécessaires pour la construction du modèle général pouvant servir de support de représentation de l'information :
    - L'uroscan est composée d'acquisitions présentant une information complémentaire. La première étape consiste à mettre en correspondance ces différentes acquisitions par des techniques de recalage. Malheureusement, le volume abdominal se déforme continuellement dans le temps. Cette mise en correspondance a été effectuée par une technique de *recalage local centrée sur le volume rénal et basée sur la maximisation de l'Information Mutuelle*.
    - Après le recalage, nous sommes en présence maintenant d'un volume où chaque voxel porte un vecteur de N valeurs correspondant respectivement aux intensités des N acquisitions de l'uroscan (N vaut 3 ou 4 dans notre cas). L'étape suivante consiste à différencier les différentes structures anatomiques à partir de ces données. Pour cela, nous avons choisi d'utiliser une technique de *classification statistique du volume vectoriel* basée sur une modélisation de la distribution des valeurs par un *mélange de Gaussiennes* à laquelle a été ajoutée une *information spatiale* lors du processus de modélisation.

Au final, le modèle général se présente sous la forme d'un volume où les différents tissus rénaux sont décrits, en chaque voxel, sous la forme d'une probabilité de présence.
Concernant la transcription en entités graphiques, il est classique de représenter les différentes structures rénales par leurs surfaces, l'utilisateur choisissant à priori la couleur de la structure et son degré de transparence.

- La représentation graphique et la manipulation des éléments graphiques. Dans cette dernière étape de l'élaboration d'un outil de visualisation, deux problématiques ont été abordées : d'une part des techniques de visualisation de volumes vectoriels et d'autre part la simplification du maillage d'une surface décrite par des facettes.
    - Techniques de visualisation d'un volume vectoriel. Si les techniques de représentation de données scalaires ont fait l'objet de nombreuses recherches dans les années 80-90, la représentation de volumes vectoriels (ou de volumes multiples) a été peu abordée dans la littérature. Notre contribution dans ce domaine a porté sur le développement de 3 outils de visualisation 3D de l'information contenu dans le modèle général défini précédemment : 1) une technique de *rendu de surface* basée sur une extraction des surfaces contenues dans le volume par Marching Cubes [2] ; une technique de *rendu de volume* par lancer de rayons où la couleur et le degré de transparence en chaque voxel sont issues d'une procédure de *décision de classe* ; et 3) une technique de *rendu de volume* par lancer de rayons où la couleur et le degré de transparence en chaque voxel sont issues d'une procédure de *composition en fonction des probabilités d'appartenance*.
    - Simplification du maillage d'une surface décrite par des facettes. Les procédures de visualisation par rendu de surface modélisent les surfaces par un grand nombre de polygones (facettes triangulaires généralement). Ce grand nombre peut être supérieur à la capacité de traitement en temps réels des processeurs graphiques actuels. Pour cela, il est courant de diminuer ce nombre à l'aide de procédures de simplifications de maillages. Une des techniques de simplification souvent employées consiste à retirer successivement des éléments de la surface (facette, arête ou point) en fonction d'un critère de coût de préservation de la surface, et ceci jusqu'à atteindre le degré de simplification désiré. Un des points clés de toute méthode de simplification est la métrique utilisée pour estimer le degré de ressemblance entre la surface de départ et la surface simplifiée. Deux métriques décrivant la géométrie globale de la surface ont été proposées, l'une basée sur les *moments géométriques de la surface* de l'objet, l'autre sur les *moments géométriques du volume* de l'objet. Ces deux métriques peuvent être intégrées dans le processus itératif de simplification.

La suite du mémoire détaille ces points dans les chapitres : chapitre 2, une technique de recalage local par maximisation de l'information mutuelle ; chapitre 3, une méthode de classification statistique d'un volume vectoriel basée sur une modélisation par un mélange de Gaussiennes avec une information spatiale ; chapitre 4, trois techniques de visualisation d'un volume vectoriel ; et chapitre 5, les deux métriques basée sur les moments géométriques et permettant une simplification d'un modèle de surface.

Chapitre 2 : Recalage local par maximisation de l'information mutuelle

La mise en correspondance de l'information complémentaire dans un même volume spatiale de l'uroscan assurerait une description beaucoup plus précise de l'anatomie spécifique du patient. Cette mise en correspondance nécessite une étape de recalage des différentes acquisitions.

Le choix d'une méthode de recalage dépend généralement de la nature des données à traiter [3]. Notre objectif et la nature des acquisitions scanner X imposent d'utiliser une technique de *recalage 3D/3D, monomodale, intra-sujet*. Dans le cas de séquences de volumes abdominaux, les tissus se déplacent et se déforment continuellement du fait de la respiration, ce qui entraîne de grandes difficultés si nous souhaitons recaler les différents volumes abdominaux entre eux. Mais notre cas clinique ne s'intéresse qu'à l'anatomie rénale. Si le volume abdominal n'est pas rigide en soit, l'hypothèse que le rein reste rigide dans le temps peut être posée, parce que, d'une part les tissus rénaux fortement vascularisés sont denses, et, d'autre part, les acquisitions sont effectuées dans un laps de temps assez court (20 à 30 minutes). Les mouvements respiratoires ou abdominaux ne déforment donc pas le rein mais le déplacent uniquement entre les acquisitions. Du fait de cette hypothèse, une technique de recalage *rigide centrée sur le volume rénal* peut être choisie.

Notre problème se décompose donc en deux sous-problèmes, une première étape *d'extraction d'une région d'intérêt centrée sur le rein* et une seconde étape de recalage rigide proprement dit. Concernant cette dernière étape, différentes techniques sont proposées dans la littérature. Pour des questions de précision, Nous avons choisi d'utiliser une technique de *recalage basée sur les intensités des voxels et maximisant l'information mutuelle* entre les différentes acquisitions.

Extraction d'une région d'intérêt centrée sur le rein.

L'objectif n'est pas ici de segmenter précisément le rein mais simplement d'extraire un volume qui contient le rein. Une simple boite englobante ne paraît pas satisfaisante car incluant trop d'information extra-rénale ; en contre partie, une détection automatique du rein est relativement difficile. Entre ces deux extrêmes, une méthode de segmentation semi-automatique paraît raisonnable. Une telle méthode, basée sur des contours actifs, a été étudiée afin d'extraire globalement le volume rénal dans un temps raisonnable. Les différentes étapes de cette segmentation sont les suivantes :

1) Sur une première coupe, le contour du rein est initialisé en sélectionnant manuellement quelques points. Ce contour s'ajuste alors automatiquement par la méthode des contours actifs. Ce contour est ensuite propagé vers la coupe suivante pour servir d'initialisation au nouveau contour actif. Cette procédure de propagation/ajustement se poursuit jusqu'à la dernière coupe. Pendant cette procédure automatique, des ajustements manuels restent possibles.
2) Les contours externes du rein ainsi extraits sont utilisés pour former une surface 3D fermée. Cette surface est remplie pour former un volume binaire.

3) Le volume est dilaté afin d'être sûr que toute l'information du rein soit incluse dans ce volume.
4) Ce volume sert alors de masque pour extraire le volume rénal en niveau de gris.

Le temps mis pour l'extraction d'un volume rénal est de l'ordre de 3 minutes pour un utilisateur non entraîné.

Recalage basé sur les intensités des voxels et maximisant l'information mutuelle.

Une méthode de recalage est généralement caractérisée par trois aspects importants : le modèle de transformation, la mesure de similarité et la méthode d'optimisation.

- Modèle de transformation. La première étape consistait à extraire les deux régions d'intérêt incluant le rein et l'échelle des acquisitions est connue. Nous avons donc à recaler les deux régions d'intérêt mises à la même échelle en millimètre. Ayant posé l'hypothèse que la forme du rein est invariante dans le temps, le recalage recherché est de type rigide avec trois paramètres de translation et trois paramètres de rotations à estimer.
- Mesure de similarité. Elle décrit numériquement le degré de ressemblance entre deux volumes. Cinq mesures de similarité ont été implémentées : 1) l'information mutuelle, 2) le coefficient de corrélation d'entropie, 3) l'information mutuelle normalisée, 4) le rapport de corrélation et 5) l'entropie de Tsallys. Ces mesures sont appliquées sur les voxels se superposant entre le volume cible après transformation et le volume de référence et sont formulées à partir de la densité de probabilité conjointe (estimée à partir de l'histogramme 2D conjoint).

  Nous avons voulu évaluer ces cinq mesures de similarité dans le cadre spécifique de notre application. Pour cela, nous nous sommes basés sur un protocole proposé par Skerl qui évalue la performance des mesures de similarité [4]. Connaissant les paramètres de la transformation exacte, le protocole choisit, selon des critères de régularité, différents jeux de paramètres dans le voisinage des paramètres idéaux. Des mesures de similarité sont effectuées sur ces jeux de paramètres. Des statistiques sont réalisées à partir de ces mesures. Elles permettent d'évaluer : la précision, le gradient moyen de variation des mesures, le nombre de minima locaux, le risque de non-convergence,…. Ce protocole est appliqué, d'une part sur des données de synthèse (un volume rénal de référence et un volume cible fabriqué à partir du volume de référence auquel une transformation connue est appliquée) et, d'autre part, sur des données réelles. Dans ce dernier cas, la transformation estimée par maximisation de l'information mutuelle est considérée comme le jeu idéal de paramètres et seules les statistiques portant sur le gradient moyen de variation des mesures, le nombre de minima locaux et le risque de non-convergence seront examinées. Dans les deux cas, données de synthèse et données réelles, l'information mutuelle et l'information mutuelle normalisée donnent les meilleurs résultats. L'information mutuelle sera donc retenue comme méthode de mesure de similarité.
- Méthode d'optimisation. Plusieurs méthodes d'optimisation existent dans la littérature : recuit simulé, Powell, downhill simplex,… Nous avons choisie cette dernière méthode car réputée rapide et assez robuste aux minima locaux. Par contre, nous nous sommes intéressés à l'influence du choix des paramètres

initiaux. Sur des données de synthèse dont la transformation est connue, nous avons tracé l'évolution de la mesure de similarité dans l'espace des paramètres. Nous constatons que des minima locaux apparaissent en fonction de certaines configurations des paramètres initiaux et des transformations à estimer. La conclusion de cette analyse est que les paramètres initiaux doivent être proches des paramètres à estimer. De ce fait, nous utilisons une première méthode de recalage utilisant les moments géométriques afin d'initialiser la méthode de maximisation de l'information mutuelle.

Deux autres points ont été pris en compte et évalués pour l'implémentation de notre technique de recalage : la construction de l'histogramme 2D conjoint (nous avons choisi de construire l'histogramme en distribuant les volumes partiels) et la résolution de cet histogramme (différents tests sur le tracé de l'évolution de la mesure de similarité dans l'espace des paramètres nous ont fait choisir une résolution de 64 bins par axe ; cette résolution donne le meilleur compromis entre temps de calcul et précision).

Résultats

Cette technique a été évaluée sur des données de synthèse et des données réelles. Les données de synthèse ont été crées de la façon suivante : un volume rénal (dont la résolution est de 0,65 mm) est choisi comme référence et un volume cible est fabriqué par l'application d'une transformation connue au volume de référence. Différents jeux de couple de données ont été réalisés à partir de paramètres de transformation tirés aléatoirement. Les paramètres estimés par notre méthode de recalage sont alors comparés aux paramètres réels. L'erreur maximale en translation entre les paramètres estimés et les paramètres réels est inférieure à 0,08 mm, l'erreur angulaire maximale en rotation est inférieure à 0,1°.

La technique de recalage appliquée sur différentes acquisitions d'un uroscan montre visuellement un bon alignement des différents volumes rénaux. Ainsi les différents volumes rénaux peuvent être fusionnés dans même référentiel spécifique au patient.

Chapitre 3 : Classification statistique d'un volume vectoriel

Les volumes recalés par la technique décrite dans le chapitre 2, présente l'information complémentaire alignée dans un même référentiel spatial. Nous sommes en présence maintenant d'un volume où chaque voxel porte un vecteur de N valeurs correspondant respectivement aux intensités des N acquisitions de l'uroscan (N vaut 3 ou 4 dans notre cas). L'étape suivante de la construction d'un modèle général des données à visualiser, consiste à différencier les différentes structures anatomiques à partir de ces données. Cette étape, que nous allons traiter dans ce chapitre, est une étape de segmentation ou de classification et plus précisément d'une technique de *classification multidimensionnelle.*

Une des particularités des volumes acquis en imagerie médicale est que les objets ne sont pas décrits de manière explicite. Les données recueillies sont issues de mesures physiques échantillonnées spatialement de manière régulière à une certaine

résolution. La valeur en chaque voxel est donc le résultat d'une combinaison complexe de mesures avec des effets de volume partiel et de bruit de mesure. Cette complexité nous conduit à choisir une technique de classification où en chaque voxel nous allons définir une probabilité de présence d'un certain tissu plutôt que décider de manière définitive qu'un certain tissu est localisé à ce voxel. Ces probabilités de présence de tissus seront alors estimées à l'aide d'une *classification statistique* de données.

Parmi les différentes méthodes de classification statistique de données scalaires, le modèle basé sur un mélange de Gaussiennes (Gaussian mixture model) est déjà utilisé en segmentation d'images médicales (d'IRM en particulier). Nous avons utilisé cette méthode de *classification statistique par mélange de Gaussiennes* et l'avons étendu au cas de données décrites par des vecteurs de valeurs.

L'inconvénient principal de cette méthode réside dans le fait qu'elle n'est basée que sur la distribution des valeurs et qu'elle ne tient compte d'aucune information spatiale. Or, l'information spatiale (et particulièrement pour un voxel donné, l'information issue de son voisinage immédiat) est une source d'information essentielle en traitement d'images ou de volume médicaux. Nous avons donc modifié la méthode de classification précédente en proposant un *modèle de mélange de Gaussiennes pondérées par le voisinage*.

Classification statistique par mélange de Gaussiennes.

Ce modèle est basé sur l'hypothèse que la distribution des valeurs observées, représentatives d'un certain tissu, suit une loi Normale $f_k(x)$ dont les paramètres sont la moyenne et la variance de la distribution. La valeur observée en chaque voxel est une combinaison linéaire des valeurs issues des distributions de chaque classe de tissu. Ainsi, si nous avons K classes de tissus et donc K distributions Normales $f_k(x)$ avec $1 \leq k \leq K$, la valeur en chaque voxel est donnée par la formule suivante : $f(x) = \sum_{k=1}^{K} \alpha_k f_k(x)$ avec α_k qui est le coefficient de mélange de la classe k. L'objectif de la classification statistique est d'estimer, à partir des données, un jeu de paramètres Θ composé des paramètres de chaque distribution $f_k(x)$ (moyenne et variance de la loi Normale) et des coefficients de mélange a_k.

La technique de la maximisation de l'espérance (Expectation Maximisation) est une méthode générale pour estimer les paramètres des distributions sous-jacentes à un jeu de données incomplets au sens du maximum de vraisemblance. Appliquée à notre cas de mixture de Gaussiennes, cette méthode permet d'identifier de manière itérative, à partir des données, les paramètres de chaque distribution et les coefficients de mélange. Cette technique itérative se décompose en deux étapes : 1) en chaque voxel x_i, la probabilité conditionnelle de chaque classe k est estimée en utilisant la règle de Bayes : $p(k|x_i,\Theta^t) = \frac{\alpha_k f_k(x_i)}{\sum_{k=1}^{K} \alpha_k f_k(x_i)}$; 2) le jeu de paramètres Θ^{t+1} à l'itération suivante est estimé à partir des $p(k|x_i,\Theta^t)$.

Cette méthode est généralement utilisée sur des données scalaires, mais elle peut être étendue à notre cas de description de données par un vecteur de valeurs : x est maintenant un vecteur de N composantes et $f_k(x)$ est une distribution normale de dimension N (la moyenne devient un vecteur de N composantes et la variance de la Gaussienne devient une matrice de covariance).

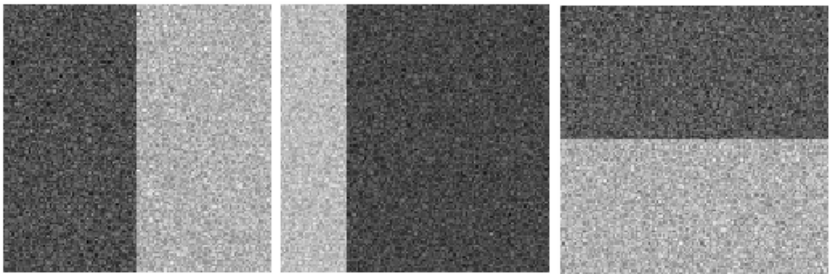
Modèle de mélange de Gaussiennes pondérées par le voisinage.

Le problème inhérent à toute méthode de classification uniquement basée sur la distribution de valeurs est que l'information spatiale n'est pas prise en compte. Or, cette information est généralement essentielle pour le traitement d'image. Nous proposons une méthode qui permet de prendre en compte l'information de voisinage lors du calcul des probabilités de classes en chaque voxel. Pour cela, nous sommes partis de l'hypothèse que les tissus sont continus et que la probabilité d'une classe k en un voxel est influencée par les probabilités de la classe k des voxels du voisinage. Nous avons donc introduit, à la probabilité d'une classe k en un voxel, une pondération qui est fonction des probabilités de la classe k des voxels du voisinage :

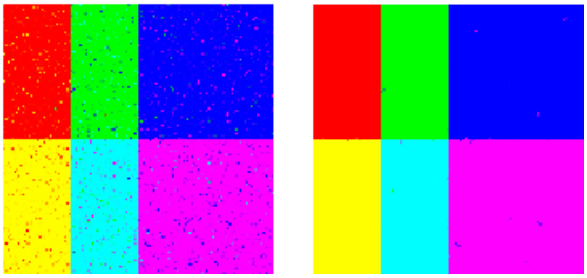
$$p\left(k|x_i,\Theta^t\right)=\frac{\alpha_k W_{ik} f_k\left(x_i\right)}{\sum_{k=1}^{K}\alpha_k W_{ik} f_k\left(x_i\right)} \text{ avec } W_{ik}=\frac{\sum_{n=1}^{|N_i|} p\left(k|x_n,\Theta^t\right)}{|N_i|}$$ calculé sur un voisinage N_i choisi. L'estimation du poids W_{ik} est intégrée dans la procédure itérative de maximisation de l'espérance.

Résultat

Cette méthode a été évaluée dans un premier temps sur une image 2D de synthèse où chaque pixel est un vecteur de 3 éléments. Chaque élément est représentatif d'une image composée de 2 régions homogènes auxquelles nous avons ajouté du bruit Gaussien.



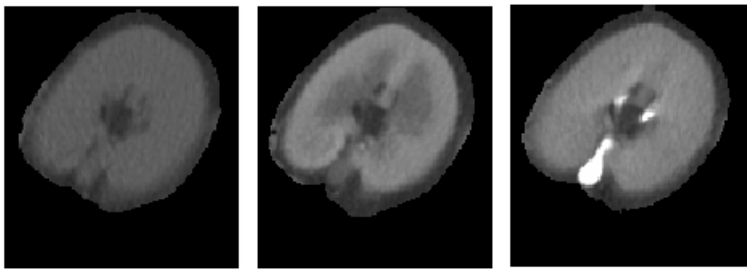
La combinaison de ces trois éléments forme une image composée de 6 régions. Les résultats de la classification sont présentés ci-dessous avec en (a) la méthode estimant le mélange de Gaussiennes basée sur les seules valeurs et en (b) lorsque la méthode est pondérée par le voisinage.



(a) (b)

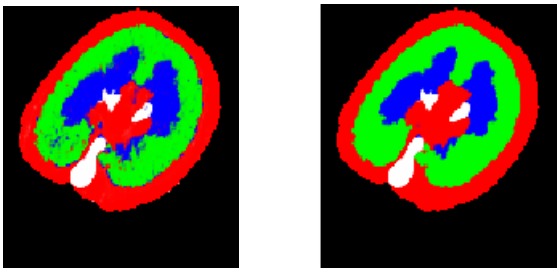
Sur ces images, la couleur attribuée aux différents pixels est représentative de la classification statistique et est obtenue par la formule suivante : $C(x_i) = \sum_{k=1}^{K} C_k \, p(k|x_i, \Theta)$ avec C_k, la couleur attribuée a priori à la classe k. Nous pouvons noter que les régions sont plus homogènes avec la méthode pondérée par le voisinage.

Nous avons ensuite appliqué les deux méthodes sur des données réelles : un volume rénal dont chaque voxel est un vecteur de 3 éléments correspondants respectivement à (a) l'acquisition avant injection, (b) à l'acquisition du stade parenchymateux et (c) à l'acquisition prise 10 minutes après l'injection du produit de contraste. Les trois composantes d'une coupe de ce volume sont représentées dans la figure suivante.



(a) (b) (c)

Nous voulons classer ce volume en 4 tissus : graisse (rouge), cortex (vert), medulla (bleu) et voie urinaires (blanc). Les résultats de la classification sont présentés ci-dessous avec en (a) la méthode estimant le mélange de Gaussiennes basée sur les seules valeurs et en (b) lorsque la méthode est pondérée par le voisinage.



(a) (b)

Nous retrouvons les mêmes caractéristiques de classification que pour les données de synthèse.

Nous constatons au final que l'introduction de l'information de voisinage dans les étapes de la classification rend cette dernière moins sensible au bruit et aux effets du volume partiel.

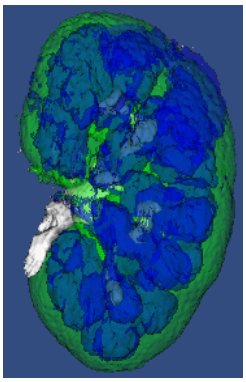
Chapitre 4 : Techniques de visualisation d'un volume vectoriel

Si les techniques de représentation de données scalaires ont fait l'objet de nombreuses recherches dans les années 80-90, la représentation d'un volume vectoriel (ou de volumes multiples) a été peu abordée dans la littérature. Or, les traitements d'images médicales proposent de plus en plus de techniques de fusion d'information multiple dans un même cadre anatomique, tant pour un but de diagnostic que d'aide au geste chirurgical, d'où un besoin croissant d'outil de visualisation de cette information multiple. Les techniques proposées jusqu'à maintenant sont basées sur des techniques de représentation de données scalaires et se caractérisent par le choix de l'étape du processus de formation de l'image où les données multiples sont fusionnées [5]. Mais dans tous ces cas, ces données multiples sont considérées comme indépendantes avant la fusion. Dans notre cas, les différentes données sont fusionnées avant le processus de visualisation. Nous allons donc étudier différentes techniques de représentation qui intègrent cette notion de données vectorielles dans le processus de visualisation.

Traditionnellement, dans le domaine médical, les techniques de représentation 3D ont été classées en deux catégories : les technique dites de "surface rendering" caractérisées par le fait que les surfaces qui seront représentées sont dans un premier temps extraites (ou segmentées) du volume et décrites typiquement sous la forme de facettes ; et les technique dites de "volume rendering" où l'image finale est formée directement à partir de l'information du volume sans segmentation préalable. Nous nous sommes intéressé à ces deux classes de technique et les avons adaptées à la visualisation de données vectorielles.

Méthode de "surface rendering"

Dans le cadre de la visualisation de volumes de données scalaire, les techniques dites de "surface rendering" sont basées sur une extraction et une approximation des surfaces à représenter. Dans le domaine médical, la technique classique consiste, à l'aide de l'algorithme du Marching Cube [2], à extraire une surface d'isovaleur et à la décrire par des facettes. Ces facettes sont ensuite traitées par les processeurs graphiques des ordinateurs. Cette technique d'extraction de surfaces n'est pas transposable directement au cas des données vectorielles car la notion d'isovaleur y est difficilement définissable. Toutefois, la technique de classification décrite dans le chapitre 3 donne des résultats homogènes avec des transitions nettes entre tissus. Il est donc possible d'utiliser la technique du Marching Cube pour chacun des tissus, d'attribuer couleur et transparence à chaque surface ainsi extraite et d'associer ces différentes surfaces pour la représentation finale.



Représentation des surfaces (cortex en vert, medulla en bleu et voies urinaires en blanc)

Méthode de "volume rendering"

Dans cette classe de méthode, l'image finale est formée directement à partir de l'information du volume sans segmentation préalable. Nous avons choisi de nous intéresser à des techniques de type lancer de rayons [6]. Dans cette technique, la procédure de calcul d'une image à partir d'un volume de données scalaires est la suivante : 1) des *rayons* sont lancés de l'écran à travers le volume ; 2) chaque rayon est *échantillonné* spatialement ; 3) la valeur recueillie à la position courante d'un échantillon sert à : 3a) déterminer une *opacité,* 3b) déterminer une certaine *couleur* en fonction du tissu et d'un *ombrage* estimé à l'aide d'un calcul de *gradient 3D* local ; 4) la couleur finale d'un rayon est obtenue par la *composition* des opacités et des couleurs le long de ce rayon.

L'opacité et la couleur en un voxel sont déterminées à partir de sa valeur par des fonctions de transfert, ce qui permet plusieurs types de représentation de l'information.

Cette procédure de lancer de rayons peut être adaptée à notre cas de données vectorielles après la résolution de certains points clés qui ont fait l'objet de nos travaux de recherche :

- le calcul de gradient 3D. La notion de gradient 3D est difficilement transposable dans le cas de données vectorielles, il nous faut donc passer par une variable intermédiaire qui reflète l'homogénéité des tissus ou au contraire la présence d'une surface entre deux tissus. En nous inspirant d'une idée émise par Drebin [7], nous avons créé une nouvelle caractéristique du volume, la densité, déterminée par la formule suivante : $D = \sum_{k=1}^{K} p_k \alpha_k$, avec K, le nombre de tissus, p_k, la probabilité de la présence du tissu k et α_k, le degré de transparence du tissu k choisi à priori par l'utilisateur. Dans le volume, les voxels d'une région homogène auront des densités proches. Ainsi, une surface (le passage d'une région à une autre) sera caractérisée par un saut de densité et l'orientation de cette surface pourra être estimée par un opérateur de gradient 3D appliqué sur ces densités.

- L'estimation de l'opacité et de la couleur en un échantillon. Dans la technique classique de volume rendering, l'opacité et la couleur sont déterminées localement à partir de la valeur par des fonctions de transfert. Ces fonctions de transfert peuvent avoir deux rôles [6] : a) d'une part, attribuer à un voxel l'opacité et la couleur qui dépendent des tissus ; b) d'autre part, rehausser les surfaces contenues

dans le volume en augmentant l'opacité des zones frontières et en la diminuant dans les zones homogènes. Dans notre cas, nous devons construire des fonctions de transfert qui déterminent l'opacité et la couleur à partir des probabilités p_k de présence des tissus. Deux méthodes ont été introduites. Ces deux méthodes essayent de résoudre à la fois le problème d'attribution d'une opacité en fonction du tissu et celui du rehaussement des surfaces.

- *La méthode de décision de classe*. L'idée est, pour un voxel donné, de décider quel est le tissu le plus probable et de lui attribuer les propriétés de ce tissu. Pour cela, les dérivés premières des probabilités sont analysées lors de la progression le long du rayon. Pour un voxel donné on lui attribue le tissu *k* qui présente la dérivée de la probabilité *f'(k)* la plus grande. Après cette décision, la couleur C_s et l'opacité α_s sont attribuées au voxel courant par :

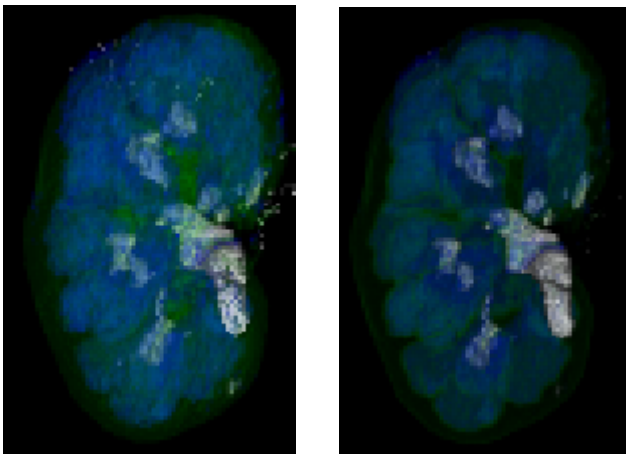
  $C_s = C_k$ et $\alpha_s = \alpha_k \, . \, f'(k)$

  En analysant cette formule, nous constatons que les volumes homogènes seront transparents (*f'(k)* proche de 0) et que seules les surfaces présenteront une certaine opacité.
- *La composition des couleurs et opacités*. L'idée est d'obtenir directement la couleur et l'opacité à partir des probabilités de présence de tissus. Pour cela, en un voxel donné, la couleur et l'opacité qui dépendent des tissus seront données par la composition des couleurs et opacités de chaque tissu selon les formules suivantes :

  $C_s = \sum_{k=1}^{K} p_k C_k$ et $\alpha_s = \sum_{k=1}^{K} p_k \alpha_k$

  Le rehaussement des surfaces se fait par une pondération des opacités en fonction du module du gradient $\mathbf{N}_s$. Pour cela, nous utiliserons le principe décrit précédemment du calcul de gradient 3D sur les densités *D*. Pour un voxel donné, le module normalisé du gradient 3D $|\mathbf{N}_s|$ sera utilisé pour pondérer l'opacité : $\alpha_s = \left(\sum_{k=1}^{K} p_k \alpha_k\right) . |\mathbf{N}_s|$

Ces deux méthodes d'estimation de l'opacité et de la couleur ont été implémentées dans le cadre du lancer de rayons.



(a) (b)

Méthode de "volume rendering" en utilisant (a): la décision de classe; (b): la composition des couleurs et opacités

Chapitre 5 : Simplification d'un modèle de surface.

Les techniques de rendus de surface sont classiquement basées sur une description des surfaces par des polygones (facettes triangulaires le plus souvent). Or les techniques qui permettent d'extraire ou de définir ces surfaces (Marching Cubes par exemple) produisent généralement une description présentant un très grand nombre de facettes. Ce grand nombre de facettes est souvent supérieur à la capacité de traitement en temps réel des processeurs graphiques actuels. Une solution naturelle à ce problème consiste à essayer de réduire le nombre total de facettes tout en préservant l'aspect global de la surface.

Une des techniques de simplification souvent employée est la technique de fusion d'éléments qui consiste à agréger successivement des éléments de la surface (facettes, arêtes ou points) en fonction d'un critère de coût de préservation de la surface, et ceci jusqu'à atteindre le degré de simplification désiré. Un des points clés de toute méthode de simplification est la métrique utilisée pour estimer le degré de ressemblance entre la surface de départ et la surface simplifiée. En effet, c'est sur cette métrique que repose le choix des éléments de surface à fusionner et donc la qualité finale de la simplification de la surface. La plupart des métriques proposées dans la littérature repose sur des critères locaux qui préservent les caractéristiques locales de la surface. Toutefois certains auteurs ont proposé des critères globaux afin de préserver plutôt la forme globale de la surface à simplifier : Park et al utilisent comme métrique de préservation, la différence entre l'aire de la surface simplifiée et l'aire de la surface de départ ; selon la même idée, Alliez et al utilisent la différence entre le volume de la surface simplifiée et celui de la surface de départ. L'information spatiale ou géométrique portée par l'aire ou le volume est toutefois assez restreinte et de bas niveau. D'autres mesures géométriques globales, telles les moments géométriques, portent plus d'information spatiale sur la forme de la surface. En se basant sur cette idée, nous proposons deux nouvelles métriques : l'une utilisant les *moments géométriques de la surface* de l'objet, l'autre utilisant les *moments géométriques du volume* de l'objet.

Moments géométriques de la surface.

Les moments géométriques d'ordre $k_1+k_2+k_3$ de la surface d'un objet P sont définis par : $m_{k_1k_2k_3}S(P)=\int_{S(P)} x^{k_1}y^{k_2}z^{k_3}dS$ où l'intégrale est effectuée sur la surface de P. Si nous désirons mesurer la similarité entre la surface de départ S de P et une surface simplifiée S' de P, nous pouvons définir comme métrique : $SMD=\left|\sum_{p=0}^{M}\sum_{q=0}^{p}\sum_{r=0}^{q}\left(m_{p-q,q-r,r}S(P)-m_{p-q,q-r,r}S'(P)\right)\right|$ où M est l'ordre maximal des moments que nous voulons utiliser.

Il est a noter que $m_{0,0,0}S(P)$ est l'aire de la surface de P et donc, si nous nous restreignons à M=0, nous retrouvons la métrique proposée par Park et al.

Pour M>0, notre métrique apporte donc plus d'information que la métrique basée sur la seule différence entre surfaces. Toutefois l'inconvénient de la métrique basée

sur les moments est qu'elle est assez longue à calculer. Toutefois, ce problème est contourné en utilisant l'algorithme simple et rapide d'estimation des moments de surfaces proposé par Tuzikov et al [8]. De plus nous démontrons que lors de l'opération de simplification, nous pouvons estimer l'évolution des moments de la surface de la forme simplifiée en ne regardant que l'impact de la modification sans avoir à recalculer les moments sur la surface entière.

Moments géométriques du volume.

Les moments géométriques d'ordre $k_1+k_2+k_3$ du volume d'un objet P sont définis par : $m_{k_1k_2k_3}V(P)=\int_P x^{k_1}y^{k_2}z^{k_3}dxdydz$ où l'intégrale est effectuée sur le volume de P. Si nous désirons mesurer la similarité entre le volume de départ V de P et le volume V' de la surface simplifiée S' de P, nous pouvons alors définir comme métrique : $VMD=\left|\sum_{p=0}^{M}\sum_{q=0}^{p}\sum_{r=0}^{q}\left(m_{p-q,q-r,r}V(P)-m_{p-q,q-r,r}V'(P)\right)\right|$ où M est l'ordre maximal des moments que nous voulons utiliser.

Comme précédemment il est à noter que $m_{0,0,0}V(P)$ est le volume de la surface de P et donc, si nous nous restreignons à M=0, nous retrouvons la métrique proposée par Alliez et al.

Comme pour les moments géométriques de surface, nous pouvons utiliser une méthode de calcul rapide des moments géométriques de volumes et nous avons démontré que lors de l'opération de simplification, nous pouvons estimer l'évolution des moments de volume de la forme simplifiée en ne regardant que l'impact de la modification sans avoir à recalculer les moments sur le volume de la surface entière.

Évaluation des métriques

Nous nous sommes inspiré du schéma d'évaluation proposé par [4] afin d'estimer la performance des deux métriques proposées.

Ce schéma consiste dans un premier temps, à choisir une méthode de simplification utilisant une métrique (méthode par suppression d'arêtes dans notre cas). Pour chaque métrique à évaluer, cette méthode est alors appliquée sur une même surface de départ pour atteindre un même degré de simplification. Le degré de ressemblance entre la surface simplifiée et la surface de départ est alors estimé en mesurant l'écart type des distances entre ces surfaces afin de comparer la performance des différentes métriques entre-elles.

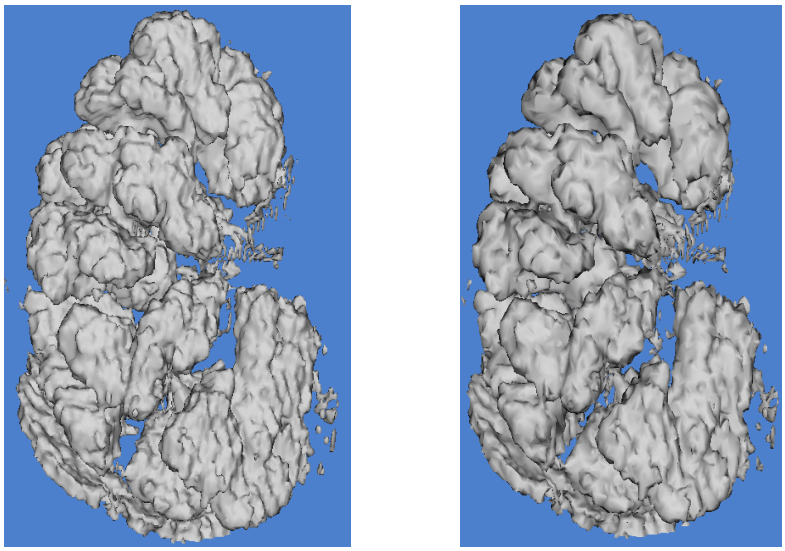
Ce protocole d'évaluation est appliqué sur différentes surfaces 3D de référence trouvées dans la littérature. Les deux métriques, moments des surfaces et moments des volumes, sont comparées entre-elles pour différents ordres de moments. De cette évaluation, plusieurs observations peuvent être émises :

a) Pour les deux types de moments (surface ou volume), les métriques utilisant des moments d'ordre 0 donnent les résultats les plus faibles. Notre hypothèse de départ que les moments d'ordre >0 portaient plus information spatiale sur les

formes que la seule aire ou le volume d'une surface est vérifiée et utilisée pour améliorer la qualité de la simplification.

b) Pour un même ordre de moment, les moments du volume donnent des résultats un peu meilleurs que les moments de la surface.

c) La métrique utilisant les moments peut également servir de méthode d'estimation de degré de ressemblance entre deux surfaces et servir à évaluer d'autres métriques (locales ou globales)



(a) Modèle original (180140 triangles) (b) 30% (54042 triangles)
Résultats de la simplification du maillage de la surface de la medulla rénale en utilisant la métrique basée sur les moments d'ordre deux du volume de l'objet : (a) maillage original, (b) maillage après simplification et ne possédant plus que 30% du nombre de facettes de départ.

Chapitre 6 : Conclusions.

Dans cette Thèse, nous avons abordé plusieurs aspects du processus de création d'un outil de visualisation appliqué au domaine biomédical. Nous sommes partis d'une problématique médicale bien précise qui est le besoin de la représentation des structures anatomiques rénale dans le cadre du planning préopératoire d'un traitement de tumeurs rénales. Le processus de création de cet outil de visualisation a été décomposé en différents sous-problèmes : le *recalage local par maximisation de l'information mutuelle* du volume rénal afin de mettre en correspondance l'information complémentaire issue de l'uroscan ; l'estimation de la probabilité de présence de tissus en chaque voxel du volume par une méthode de *classification statistique basée sur un modèle de mélange de Gaussiennes pondérées par le voisinage* ; la représentation des structures anatomique en se basant sur technique de *visualisation à partir de ces probabilités de présence de tissus* ; et la simplification des maillages à l'aide d'une *métrique utilisant les moments géométriques*.

Les différentes solutions proposées lors de notre travail de Thèse ont été développées dans le cadre bien spécifique de notre application médicale : la

représentation des structures anatomiques rénale après acquisition par uroscan. Par contre, elles sont suffisamment génériques pour être utilisées ou adaptées à d'autres organes ou à d'autres applications médicales, soit en tant que solution complète de visualisation, soit de manière indépendante afin de répondre à des problématiques particulières.

Toutefois, ces travaux n'offrent que quelques éléments de réponses à l'élaboration de la chaîne de traitement. Plusieurs des points présentés dans ce mémoire restent à être améliorés, essentiellement dans l'automatisation des tâches (extraction d'une région d'intérêt centrée sur le rein par exemple), dans le degré de précision (classification des réseaux vasculaires ; une solution de classification par étapes successives est envisagée) et dans l'optimisation et l'accélération de certains traitements (calcul de l'image finale en volume rendering).

Références bibliographiques du résumé

[1] J.-L. Dillenseger, "Visualisation Scientifique en médecine. Application à la visualisation de l'anatomie et à la visualisation en épileptologie clinique," vol. HDR: Université de Rennes 1, 2003.

[2] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface reconstruction algorithm," *Computer Graphics*, vol. 21, pp. 163-169, 1987.

[3] J. B. Maintz and M. A. Viergever, "A survey of medical image registration," *Med Image Anal*, vol. 2, pp. 1-36, 1998.

[4] D. Skerl, B. Likar, and F. Pernus, "A protocol for evaluation of similarity measures for rigid registration," *IEEE Trans Med Imaging*, vol. 25, pp. 779-91, 2006.

[5] W. Cai and G. Sakas, "Data Intermixing and Multi-volume Rendering," *Computer Graphics Forum*, vol. 18, pp. 359-368, 1999.

[6] M. Levoy, "Display of surfaces from volume data," *IEEE Comp Graph & Appl*, vol. 8, pp. 29-37, 1988.

[7] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," presented at SIGGRAPH'88, Atlanta, GA, 1988.

[8] A. V. Tuzikov, S. A. Sheynin, and P. V. Vasiliev, "Computation of volume and surface body moments," *Pattern Recognition*, vol. 36, pp. 2521-2529, 2003.

Research of New Medical Volume Visualization Methods: Application in the Kidney Preoperative Planning System

Contents

Chapter 1: Introduction

Scientific visualization is currently a very active and vital area of research, teaching and development. Since vision dominates our sensory input, strong efforts have been made to bring the power of mathematical abstraction and modeling to our psycho-visual system through the media of computer graphics. This dissertation makes focuses on some of these problems in visualization and finally explores a specific application of the visualization techniques in a preoperative planning system for kidney surgery.

This introductory chapter introduces a brief description of scientific visualization especially in medical area. The visualization application background for kidney preoperative planning is also introduced. Then the main research contributions are presented and finally the outline of the dissertation is given.

1.1. Scientific visualization

Scientific visualization contains a wide range of aspects and algorithms. For an extensive overview and classification of scientific visualization techniques, the reader could refer to Brodlie et al. [1] or Domik [2]. In this section, we aim only to provide some basic concepts in order to better understand the scientific visualization and to express our work within this visualization framework.

1.1.1. Scientific visualization definitions

The human's vision, by its psycho-visual and cognitive faculties, represents a natural way for observing and understanding phenomena surrounding the individual. The basic idea of the scientific visualization is to exploit the inherent properties of the human vision for the analysis of different kinds of data, information or knowledge produced by scientific or other communities.

Scientists define their needs to explore scientific data and thus drive the quest for some visual exploration tools. Scientific visualization provides concepts, methods and tools from existing disciplines to best use human abilities and computer algorithms for the display of scientific data. It applies the representation of graphical data for gaining understanding and insight of the data. Sometimes it is referred as visual data analysis. This enables the researcher to gain insight into the system in ways which were previously impossible.

The research field of scientific visualization can be defined by all of the techniques which are used to explore, extract or transcode the data, phenomena or complex structures with many variables and high dimensions together into a graphical form understandable by the human's psycho-visual and cerebral system in order to enable the observer to construct a mental model of the underlying processes which are contained in the complex scene.

From the previous definition, it seems that the scientific visualization only concerns the information transcription techniques. But in fact, the visualization process is more than information transcription. Without any observer's cognitive and psycho-visual aspects, this transcription cannot make any sense. The mechanisms how the observer will understand the final visual form should also be taken into account during the information transcription procedure. This leads to the integration of the

observer into the visualization process. The global visualization process is described in Fig. 1.1.

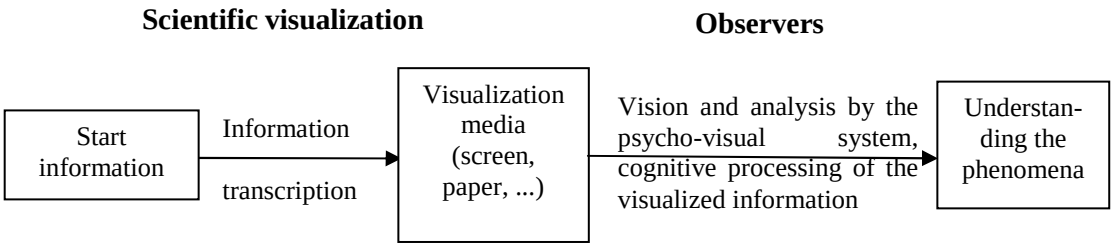


Figure 1.1: The global procedure of information processing by scientific visualization

1.1.2. Scientific visualization goals

Scientific data can be provided by various sources, including measurements or the results of scientific computations or simulations performed on supercomputers. However, data do not become useful until some (or all) of the information they carry is extracted. The goal of scientific visualization is to provide concepts, methods and tools to create expressive and effective visual representations from scientific data. Such visual representations improve the understanding of physical processes, mathematical concepts and other quantifiable phenomena expressed in the data [2]. So that scientific visualization is expected to enhance and increase scientific productivity.

It is important to differentiate scientific visualization and computer graphics. Computer graphics only concerns the techniques which can transfer graphic elements to an understandable way on the screen. In scientific visualization, we seek also to understand the data and convert it to graphic elements. A scientific visualization tool is often more than the visual representations themselves. Quantitative analysis of data, such as statistical analysis, image and signal processing techniques are often associated in order to explore more information inherent in scientific data. So that scientific visualization involves research in human perception, computer graphics, image processing, high performance computing, and other areas.

1.1.3. Visualization of medical data

Three dimensional (3D) volumetric data obtained from medical scanning is in abundance today. These data are usually obtained using scanning modalities such as computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound, etc. The acquisitions contain information about the internal anatomical structures or the physiology of the patient. With the increasing acquisitions in medical area, more and more energies are put into medical visualization research. It becomes a successful application area of scientific visualization, from the computer-assisted diagnose to computer-assisted therapy. Analyzing and illustrating the information by scientific visualization techniques makes much sense for both diagnostic and therapy usage.

The visualization of medical data can be classified into three levels: *illustrative visualization*, *investigative visualization* and *imitative visualization* [3]:

– Illustrative visualization mainly concerns the extraction of information and its presentation. According to the image spatial dimensions, the visualization can be divided into three cases: 1D displays (real-time displays, e.g. cardiogram display), 2D displays (slices, contours etc.), 3D displays (surface rendering, volume rendering etc.). Except the 1D displays (very simple), this kind of visualization focuses more on the accuracy and quality; speed is less important so that low interactivity is also acceptable.

– Investigative visualization focuses on explorative aspects and attempts to provide more information than the apparently revealed by the data. This class of visualization techniques is put into forward with the development of various medical imaging techniques (MRI, PET etc.). Several techniques are included in the investigative visualization: multimodality fusion, stereoscopic visualization, navigational visualization (e.g. virtual endoscopy) etc. The speed and interactivity are vital aspects for this kind of visualization.

– Imitative visualization attempts to imitate visual perception (virtual reality) or function (simulate and modeling). In addition, the augmented reality (e.g. intraoperative image fusion with real world objects) also belongs to this kind of visualization. Comparing to the two former visualization classes, this visualization level is higher because the user immerses in the visualization process.

Among these three levels of medical visualization, illustrative visualization has been investigated thoroughly for a long time [5-7]. Investigative visualization involves more medical image analysis techniques and because of its various applications, it becomes one of the hottest research topic in recent medical visualization research area [8-10]. Imitative visualization is the highest level, besides image analysis and visualization techniques, it also acquires the modeling techniques for the simulation of visual perception or even sensory perception [11]. The two former visualization levels can be the foundation of the research of imitative visualization.

1.2. A general medical data visualization framework

Scientific visualization is widely used in the medical field. The variety of medical data encourages the rapid development of visualization techniques application in medical area. Many medical data visualization tools have been developed for specific applications. Thinking about the generality of the design process of a medical visualization tool, in our opinion, it must follow some specific rules in order to be more efficient. Here we will present the key points which have to be taken into account during this design [12].

Overall, there are two strategies that can be followed for the design. On the one hand, the observer can use a general visualization software (e.g. AVS [13]). Such packages are either developed by researchers majored on visualization, or extended from an application of a particular area. They offer a variety of processing and representation models. The role of the users is then to choose by themselves and combine the different tools which seem the most relevant to their specific application. The other possibility is to develop a tool directly dedicated to the user's specific area. This solution requires a close cooperation between the end user and the visualization researcher in order to better define the needs and methods of representation.

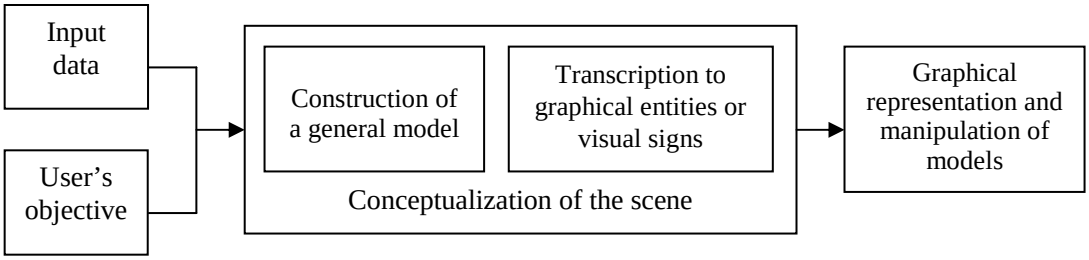


Figure 1.2: The design aspects of a visualization tool

In both cases, no matter it is done in an intuitive or more formal way, the development of a tool for medical visualization is much more than the graphical representation. There are three major points that should be taken into account (Fig. 1.2):

- The cognition about the input medical data (dimension, nature and spatial topology of the data, etc.) and the user's objectives (interest on certain variables, correlation between data, etc.).
- The conceptualization of the scene or transcription of the information contained in the data into a graphical form [1, 14]. This step can be divided into two parts [1]: 1) the construction of a general model from input data for the representation of the phenomena; 2) the transcription of the model into graphical entities or visual signs.
- The graphical representation and the manipulation of the graphic elements according to the user's objectives. This last point concerns mainly the computer graphics aspects. It is only one of the main steps of the visualization pipeline even if some authors reduce the scientific visualization into this unique aspect.

These three steps define the basic structure of scientific visualization and should be taken into account during each design of a visualization tool.

1.2.1. The user's objective and the input data

These two aspects, although they are outside of the design of the visualization tool, are the key points that will play an important role for the conception of the visualization strategy and its design.

The user's objective. The first step of the development of a visualization tool should be to interview the prospective users on different aspects, such as the data they want to analyze and how they operate or wish to exploit the data. This step is already a real analysis of the information to be transcribed by the visualization process. It integrates the scientific knowledge (medical in our case), the specific user's requests (his motivations, his expected objectives, and even his hidden objectives), the habitual practice in the area (the accustomed procedures, the conventions of the field, etc.).

The input data. If the goal of visualization research is to transform data into a perceptually efficient visual format, and if we are making statements about some visualization generality, we must be able to say something about the types of data that can exist for us to visualize [15]. The analysis of the data forms the basis of any information transcription stages. The input data can be totally different depending on

the field and even the scientific application in which they originated. But all this information shares common properties which can be classified and then transcribed in a visual form. The properties can be classified into four constituents [16]: the differentiation between invariants and variables; the number and nature of the data components; the organization level of the data components (qualitative vs. quantitative information, spatial dimension and topology, temporal organization); and finally the resolution of the components. For a specific data, the analysis of these constituents is essential because it enables us to choose some appropriate graphical encoding which express the same properties.

1.2.2. Conceptualization of the scene

The conceptualization of the scene is the central part of the visualization process. This step consists in the transcription of the information contained in the input data to a schematic and graphical form.

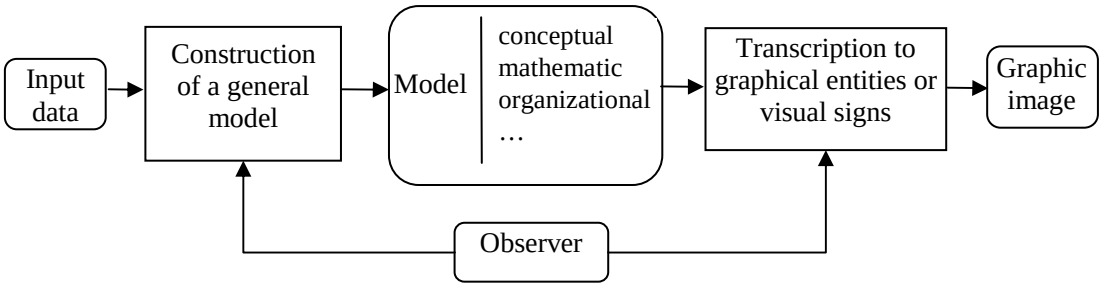


Figure 1.3: The conceptualization of the scene

Brodlie [1] separated the process of conceptualizing the scene into two stages (Fig. 1.3): one is the construction of a general model and the other is the transcription of this model to graphic entities or visual signs.

1.2.2.1. Construction of a general model

The information to be visualized can be composed of data that have relatively varied natures, locations and temporal behaviors because they are measured by different sources. The fusion of these data will require first a spatial and temporal alignment (or *registration*). The distribution of the information all over the space or time requires some *interpolation* or extrapolation techniques. Useful information can be indistinct or hidden so that *segmentation* techniques are often necessary to extract it. Another stage concerns the modeling or approximation of the information by a continuous function (for a number of techniques, the modeling or approximation step is a part of the segmentation process, but their nature is different).

Recall that the ultimate goal of these operations is to create a model of the information entities from the data. This model can be quite varied: a mathematical model, an organization model of the data, etc. This model will be used as a general support at the stage of graphic transcription. We can see that this stage of the general model construction makes extensive use of classical image analysis techniques. The following references can give a starting point for the appropriate choice of these techniques: registration [8, 10, 17], interpolation [18], segmentation [19, 20], modeling [21].

1.2.2.2. Transcription of the information to graphical cues

The models help to describe the information contained in the input data in general terms. The next logical step is to transcribe these models to graphical entities as relevant as possible. The description and classification of graphical entities have been referenced [14, 16, 22, 23].

The choice of the best suited visual variable remains delicate. The generic visualization software generally offers the user a graphic coding palette. Therefore, the user chooses a coding based on his intuition, affinities, etc. In contrast, dedicated software offer a coding which supposedly is tailored to the problem. In all cases, certain cautions must be taken when allocating a graphic code to the information components [16].

1.2.3. Graphical representation techniques

In the last stage, the visual variables are represented on the screen by computer graphics techniques [24]. In this section, we are more interested in the 3D representation forms. The objective of the 3D visualization is to represent the information which initially is three-dimensional on a picture plane (the screen). This process is the reverse of the human perception process which makes a mental reconstruction of 3D volumes from 2D projections collected by the eyes.

In medical area, the information to be visualized is often described by 3D volumetric data. Sakas [25] analyzed that the trends in medical imaging is going from 2D to 3D in recent years because the imaging procedures are being used not only by diagnosticians (usually radiologists) but also increasingly by surgeons during interventional procedures (e.g. navigation, guiding intervention, controlling therapy, etc.). This doesn't mean that slices (2D) will be totally replaced. However, the focus of future applications will be shifted from 2D to 3D.

Here we focus on the 3D rendering techniques which are used to visualize 3D reconstructions of organs. These techniques can be classified into two categories [4, 7]: *surface rendering* and *direct volume rendering* (DVR). Surface rendering involves the extraction of surface primitives from the input data, followed by projection of the extracted surface onto a 2D image. The final image quality depends on the extraction and information reduces to only surfaces. Direct volume rendering generates the 2D projections directly from the dataset. It does this by projecting the entire dataset onto the 2D image. One disadvantage of using DVR methods is that the entire dataset must be traversed each time an image is rendered. The two techniques can be combined together by *a hybrid rendering technique* when it is desirable to add geometric objects to a volumetric scene [26].

1.3. An application example: visualization for kidney surgery preoperative planning

For a dedicated application, the general visualization framework can be specified. Fig. 1.2 illustrates the general design aspects of a visualization tool. For the special kidney surgery preoperative planning visualization system, the design aspects are specified in Fig. 1.4. The detail explanation of these aspects will be introduced in this section.

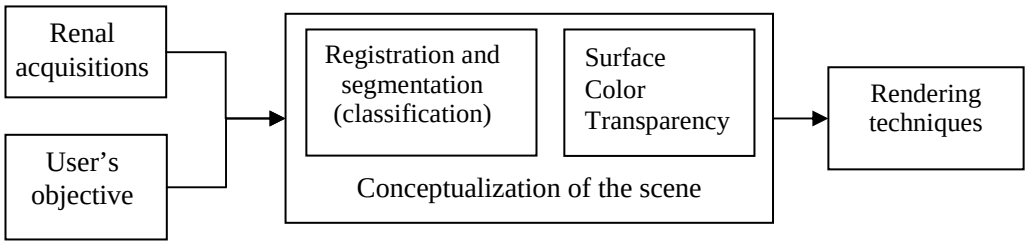


Figure 1.4: General framework of a visualization tool for the kidney surgery preoperative planning

1.3.1. Renal acquisitions and user's objective

Renal cancer represents 2~3% of whole cancers and is the third most frequent in urologic cancer. If renal tumors are detected precociously, they are usually less than 4 cm so that a nephron sparring surgery can be considered through several methods like open or celioscopic surgery, even also radiofrequency treatments. In all of these cases the surgeon needs to establish his treatment planning so that it is necessary to know the patient specific anatomy and more particularly the relations between the tumor, the vascular trees and urinary tract.

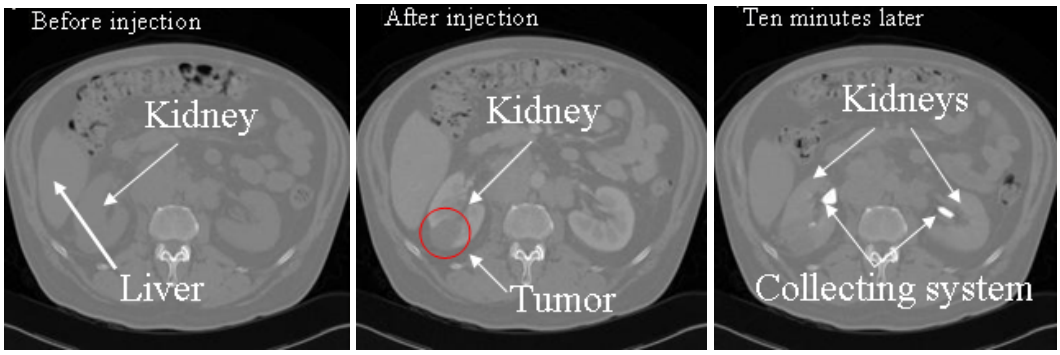


Figure 1.5: One slice of the acquisitions. From left to right: (a) slice without injection; (b) slice with vascular system and parenchyma enhanced; (c) slice with collecting system enhanced.

The CT uroscan is the classical clinical preoperative examination. It consists of three to four time spaced 3D acquisitions (Fig. 1.5), which give complementary information about the kidney anatomy. The first acquisition is realized without injection of contrast agent and informs the surgeon about intern morphology of the patient. Just after a contrast medium injection, one or two acquisitions[1] are taken, which reveal the renal vascular systems and the renal parenchyma and also give information about the nature and the location of the renal carcinoma. About ten minutes later on the last acquisition the collecting system is enhanced.

The first step of the kidney preoperative planning is to be aware about the anatomical structure of the patient's kidneys. The acquisitions mentioned above give relatively complementary information about the kidney so that it is possible to

[1] If two acquisitions are performed, the first one obtained just after the contrast medium injection reveals the renal arterial system; the second one obtained just a time later presents the venous vascularization and the renal parenchyma.

represent the patient's specific kidney anatomical structure to the surgeon by visualization techniques.

1.3.2. Conceptualization of the scene

As introduced in the general medical visualization framework, the conceptualization of the scene is separated to two stages: the construction of a general model and the transcription of this model to graphical cues.

For the specific application (kidney preoperative planning), the construction of a general model can consist of two special techniques: *registration* and *segmentation*:

- The three to four renal acquisitions at different contrast medium time give the complementary information about the patient's kidney anatomical structure. In order to integrate this information within a unique spatial volume, it is necessary to bring the different acquisitions into spatial alignment, which is called *registration*.
- After the registration, complementary information is aligned on the same spatial reference. We will use this complementary information to delineate the several renal structures. The anatomical structure information is implied in the spatial aligned acquisitions. In order to construct a model to describe this anatomical information, we should identify the tissues contained in the acquisitions by applying a suitable *segmentation* method. If this segmentation process is realized by a classifier, it is also called *classification*.

The segmentation (classification) method will give out the distribution information of the tissues in the acquisitions. The transcription of this model to graphical cues is to allocate a graphic code to the information components. Before the allocating step, we must be aware of the interested issues in the model. In our case, for illustrating the anatomical structures, the tissue *surfaces* are more important than their inside information so that we try to show the surfaces in the final image. In order to separate different tissues in the final image, we assign a *color* to each tissue. If we see from one direction, the front tissue will cover the back tissue because they are not at the same depth level. So that semi-transparent illustration is acquired to show all the tissues together. To achieve this goal, we assign a *transparency* value to each tissue.

1.3.3. Rendering techniques

The graphical representation techniques in the general visualization framework are specified to 3D rendering techniques in our case because we seek to visualize the 3D constructions of the patient's kidney.

There are two kinds of rendering techniques for the 3D visualization: *direct volume rendering* (DVR) and *surface rendering*. Direct volume rendering directly compute the final image from the original acquisitions and their properties. Depending on some algorithmic characteristics, DVR can either represent transparent surfaces or some other volume properties (X-rays similar composition). It can generate high quality images but the entire dataset must be traversed each time an image is rendered, which will cause low interaction ability. Surface rendering requires the extraction of surface primitives from the input data. The final image quality and the rendering speed mainly depend on the result of the surface construction. Sometimes the constructed surface model can contain too many surface primitives (e.g. triangles) to real-time rendering, so that a *mesh simplification* method can be applied before the projection of the extracted surface onto screen.

1.4. Contributions of our work

The research contribution of this dissertation focuses on the essential elements of a scientific visualization tool. In particular, the research area falls into the application of the kidney preoperative information review. The essential elements for the review of anatomical structure are achieved and also some additional visualization techniques are finally discussed. The individual contributions are introduced in the following chapters and they are summarized as follows:

1) A kidney-centered registration method is proposed and realized by local mutual information (MI) maximization. Kidney volumes are extracted and the registration is performed between the extracted kidneys instead of the whole volumes. The corresponding paper has been published in the international conference *IEEE EMBC* '07 [27] and the *Journal of applied sciences* (Chinese) [28].

2) A neighborhood weighted Gaussian mixture model is proposed. For the classification of the vectorial volume, we choose the statistical classification method based on Gaussian mixture model in order to acquire the material probabilities on each voxel. But this method relies only on the intensity distributions without any spatial information, which will lead to misclassification on boundaries with partial volume effects (PVE) and inhomogeneous regions with noise. In order to solve this problem we propose a neighborhood weighted solution. The proposed model is that the voxels' intensity vectors follow the Gaussian mixture distribution and that the classes distributions on each voxel are affected by its neighbors' class probability distributions. The corresponding paper has been published in the international conference *IEEE EMBC* '08 [29].

3) A new visualization method for visualizing the spatial aligned volume data is proposed. This method first intermix the several volumes at the earliest stage (acquisition level intermixing). This intermixing is realized through the neighborhood weighted Gaussian mixture model based classification method we proposed. After this data intermixing, several possible rendering techniques that can be adapted to this situation are presented and compared in this dissertation. The corresponding paper is in preparation.

4) Two new mesh simplification metrics based on surface moments and volume moments are proposed, which take the difference between the moments defined by the original mesh and those of the simplified mesh as the objective function. Comparing to most of the other mesh simplification metrics, these two proposed metrics are based on the object's global features instead of local properties. The corresponding paper has been published in the international journal *Computers & Graphics* [30].

1.5. Dissertation organization

This dissertation focuses on some issues about the scientific visualization and especially in the application of preoperative kidney information review. The summary of the chapters are listed as follows:

Chapter 1 first introduces the concept of scientific visualization and then summarizes the stages and elements for the design of a general scientific visualization tool. The kidney preoperative planning visualization is considered as the particular application for the scientific visualization techniques. Finally, the contributions and organization of this dissertation are given.

Chapter 2 proposed a kidney-centered registration method. In the range of organ-centered registration method, Dalen et al. [31] proposed a liver registration method. We implemented the similar idea but the application organ turns to kidney. The registration metrics are evaluated by an optimization independent protocol proposed by Skerl et al. [32] and MI based metrics are chosen according to the evaluation results. The experimental results are given and demonstrate the effectiveness of the proposed method.

Chapter 3 introduces the statistical classification method for the vectorial volume which is gotten after the registration. Gaussian mixture model is one of the most often used method for statistical classification and it has the ability to deal with vectorial volume. But it relies only on the intensity distributions without any spatial information, which will lead a misclassification on inhomogeneous regions with noise or on partial volume boundaries. In order to solve this problem, we propose a neighborhood weighted Gaussian mixture model in this chapter and give out the experimental results on both synthetic and real data.

Chapter 4 describes the methods for visualizing the classified vectorial volume. Both surface rendering and volume rendering techniques are described and implemented. For volume rendering based methods, two transfer function design methods are proposed and implemented for distinguish the classified materials. Finally, the experimental results are given for the comparison of different rendering methods.

Chapter 5 introduces the mesh simplification algorithms, which is one solution to speedup the surface rendering. The simplification metric is a key issue for the simplification algorithm. Two new simplification metrics based on surface moments and volume moments are proposed. The experimental results are given and the comparison with some known algorithms is also given out.

Chapter 6 summarizes the conclusions of this dissertation and outlines the ideas for the future work.

References

[1] K. W. Brodlie, L. Carpenter, R. A. Earnshaw, and J. R. Gallop, *Scientific visualization: techniques and applications*. NY, USA: Springer-Verlag 1992.

[2] G. O. Domik, "Scientific visualization: an introduction," Internal report, University of Colorado at Boulder, November 28 1995.

[3] K. Rehm, "Visualization in medical imaging," lecture notes, http://www.neurovia.umn.edu/home/kelly/Course_notes/BPHY8148/2002/Visualization_lec6.pdf, 2002.

[4] M. Levoy, "Display of surfaces from volume data," *IEEE Comp Graph & Appl*, vol. 8, pp. 29-37, 1988.

[5] M. Levoy, "Efficient ray tracing of volume data," *ACM Trans. on Graphics*, vol. 9, pp. 245-261, 1990.

[6] T. T. Elvins, "A survey of algorithms for volume visualization," presented at SIGGRAPH '92, Chicago, IL, 1992, pp. 194-201.

[7] J. Fitzpatrick, D. L. G. Hill, and C. J. Maurer, "Image registration," in *Handbook of Medical Imaging*, vol. 2, *Medical Image Processing and Analysis*. Bellingham, WA, 2000.

[8] C. L. Kay and H. A. Evangelou, "A review of the technical and clinical aspects of virtual endoscopy," *Endoscopy*, vol. 28, pp. 768-775, 1996.

[9] B. F. Hutton and M. Braun, "Software for image registration: Algorithms, accuracy, efficacy," *Seminars in Nuclear Medicine*, vol. 33, pp. 180-192, 2003.

[10] M. Krijn, P. M. G. Emmelkamp, R. Biemond, C. de Wilde de Ligny, M. J. Schuemie, and C. A. P. G. van der Mast, "Treatment of acrophobia in virtual reality: The role of immersion and presence," *Behaviour Research and Therapy*, vol. 42, pp. 229-239, 2004.

[11] J.-L. Dillenseger, "Visualisation scientifique en médecine. Application à la visualisation de l'anatomie et à la visualisation en épileptologie clinique," vol. HDR: Université de Rennes 1, 2003.

[12] C. Upson, T. Faulhaber, D. Kamins, D. Laidlaw, D. Schlegel, J. Vroom, R. Gurwitz, and A. v. Dam, "The application visualization system: a computational environment for scientific visualization," *IEEE Comp Graph & Appl*, vol. 9, pp. 30-42, 1989.

[13] P. R. Keller and M. M. Keller, *Visual cues: practical data visualization*. Los Alamitos: IEEE Computer Society Press, 1993.

[14] C. Ware, *Information visualization: perception for design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000.

[15] J. Bertin, *Semiology of graphics: diagrams, networks, maps*, 3rd ed. Paris: EHESS, 1988.

[16] J. B. Maintz and M. A. Viergever, "A survey of medical image registration," *Med Image Anal*, vol. 2, pp. 1-36, 1998.

[17] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Trans Med Imaging*, vol. 19, pp. 739-58, 2000.

[18] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, pp. 1277-1294, 1993.

[19] J. K. Udupa, V. R. LeBlanc, Y. Zhuge, C. Imielinska, H. Schmidt, L. M. Currie, B. E. Hirsch, and J. Woodburn, "A framework for evaluating image segmentation algorithms," *Computerized Medical Imaging and Graphics*, vol. 30, pp. 75-87, 2006.

[20] R. Mencl and H. Muller, "Interpolation and approximation of surfaces from three-dimensional scattered data points," presented at State of the Art Reports, Proc. of Eurographics'98, Lisbon, Portugal, 1998, pp. 51-67.

[21] E. Tufte, *Visual display of quantitative information*: Graphics press, 1987.

[22] E. Tufte, *Envisioning information*: Graphics press, 1990.

[23] A. Watt and F. Policarpo, *The computer image*: Addison Wesley, 1998.

[24] G. Sakas, "Trends in medical imaging: from 2D to 3D," *Computers & Graphics*, vol. 26, pp. 577-587, 2002.

[25] C. Barillot, "Surface and volume rendering techniques to display 3-D data," *IEEE EMB Magazine*, vol. 12, pp. 111-119, 1993.

[26] M. Levoy, "A hybrid ray tracer for rendering polygon and volume data," *IEEE Comp Graph & Appl*, vol. 10, pp. 33-40, 1990.

[27] H. Tang, J. L. Dillenseger, and L. M. Luo, "Intra subject 3D/3D kidney registration using local mutual information maximization," presented at Conf of IEEE EMBS (EMBC'07), Lyon, France, 2007, pp. 6380-83.

[28] H. Tang, J. L. Dillenseger, and L. M. Luo, "Registration of kidney in preoperation planning," *Journal of Applied Sciences (Chinese)*, vol. 25, pp. 370-376, 2007.

[29] H. Tang, J. L. Dillenseger, and L. M. Luo, "A vectorial image classification method based on neighborhood weighted Gaussian mixture model," presented at Conf of IEEE EMBS (EMBC'08), Vancouver, Canada, 2008.

[30] H. Tang, H. Z. Shu, J. L. Dillenseger, X. D. Bao, and L. M. Luo, "Moment-based metrics for mesh simplification," *Computers & Graphics*, vol. 31, pp. 710-718, 2007.

[31] J. A. v. Dalen, W. Vogel, H. J. Huisman, W. J. G. Oyen, G. J. Jager, and N. Karssemeijer, "Accuracy of rigid CT-FDG-PET image registration of the liver," *Phys. Med. Biol.*, vol. 49, pp. 5393-5405, 2004.

[32] D. Skerl, B. Likar, and F. Pernus, "A protocol for evaluation of similarity measures for rigid registration," *IEEE Trans Med Imaging*, vol. 25, pp. 779-91, 2006.

Chapter 2: Registration using local mutual information maximization

As introduced in chapter 1, the three to four time-spaced 3D acquisitions of the uroscan give complementary information about the kidney anatomy. It is useful for the surgeon to integrate this information within a unique spatial volume. The first step in this integration process is to bring the different acquisitions into spatial alignment, which is referred as registration.

Because of the respiration, the tissues shift continuously so that it is difficult to do registration between the whole abdomen acquisitions. Fortunately, we only need to focus on the kidneys. Although the abdomen is definitely not rigid, the kidney can be considered rigid because its tissues are dense and the time distance between these acquisitions is very short. The nature of the CT acquisitions leads us to decide for a 3D/3D, mono-modal, intra subject registration technique [1]. We suppose that the kidney shape is not deformed during the acquisition, even during the respiratory movements. This hypothesis leads us to choose a rigid kidney-centered registration technique.

In order to realize this technique, a local mutual information (MI) maximization registration method is proposed in this chapter. The kidneys are first extracted from the abdomen volumes. Then we evaluate several registration metrics by an optimization independent protocol proposed by Skerl et al. [2] and MI based metrics are chosen according to the evaluation results. Finally, the registration between the kidneys is implemented by maximizing the MI between them. The experimental results demonstrate that this method is effective.

2.1. Introduction

Since information from the three to four uroscan images acquired in the renal examination process is of a complementary nature, proper integration of useful data obtained from the separate images is often desired. A first step in this integration process is to bring the images into spatial alignment, which is called *registration*. The registration is to find transformations that relate spatial information conveyed in one image to that in another. Two analysis stages are taken into account in order to find an appropriate registration method: analyzing *the nature of the acquisitions* to be registered and finding out *a suitable registration method*.

The nature of the acquisitions

Maintz and Viergever [1] presented a survey of medical image registration techniques and classified them from several aspects. In this section we will only talk about the method classification areas which are related to our situation. According to this survey, the analysis of the acquisitions nature can be done from the three points of view: *spatial dimensions, involved modalities* and *subject*. According to the image spatial dimensions, the registration can be divided into three cases: 2D/2D, 2D/3D and 3D/3D, in which 3D/3D registration normally applies to the registration of two tomographic datasets, which is suitable for our case. The four uroscan images are all CT acquisitions so that our case belongs to the *monomodal* applications. Recall that

our goal is to bring the kidney images acquired from the four scan stages into spatial alignment. The images to be registered are from the same patient so that the registration is *intrasubject* registration from the subject aspect. In conclusion, the nature of these CT acquisitions leads us to decide for a 3D/3D, *monomodal*, *intrasubject* registration technique.

Registration methods

The registration methods can be classified into three categories: *point-based methods*, *surface based methods* and *intensity-based methods* [3]. *Point-based methods* first identify some fiducial points (features) from the pair of images. The transformation that aligns the corresponding fiducial points will then interpolate the mapping from one image to another. The location of these fiducial points can be based on interactive visual identification of anatomical landmarks, such as the junction of two linear structures (e.g. the central sulcus with the midline of the brain) or the intersection of a linear structure with a surface (e.g. the junction of septa in an air sinus) etc. Alternatively, the feature can be a marker attached to the anatomy and designed to be accurately localized by means of automatic algorithms. *Surface-based methods* involve determining corresponding surfaces in different images and computing the transformation that best aligns these surfaces. For image-to-image registration, the skin boundary surface and the outer cranial surface are frequently used. The surface representation can be simply a point set (i.e., a collection of points on the surface), a faceted surface (e.g., triangle set), an implicit surface, or a parametric surface (e.g., B-spline surface). *Intensity-based methods* involve calculating a transformation between two images using the pixel or voxel values only. The registration transformation is determined by iteratively optimizing some similarity measure calculated from all pixel or voxel values or a subset of voxels.

Comparing the three kinds of registration methods, intensity-based algorithms require much less amount of preprocessing and user-interaction than point-based and surface-based methods so that they have become the most widely used registration methods. Because of the need of feature location (point-based) or pre-extraction (surface-based), the point-based and surface-based methods require a great degree of user interaction and have typically exhibited lower accuracy than the intensity-based methods. So that intensity based registration method is applied for our situation.

Intensity-based registration method

An intensity-based registration algorithm can be generally characterized by three main components: *the transformation model*, *the similarity measure* and *the optimization method*. The optimization method finds the parameters which can get a transformation that maximize the similarity measure.

The transformation model specifies the mapping from one volume space to another volume space. The transformations are often partitioned into rigid and non-rigid ones with the latter transformations further divided into many subsets. The non-rigid transformations are far more complex and varied than rigid ones. Although the abdomen is definitely not rigid, the kidney can be considered rigid because its tissues are dense and the time distance between these acquisitions is very short. We supposed that the kidney shape is not deformed during the acquisition, even during the respiratory movements. If the kidneys are extracted from the input acquisitions, the

transformation for the registration between the kidneys can be considered rigid, so that the kidneys in these acquisitions should be firstly extracted. Lin et al. [4] indicated that the segmentation methods are relatively less focused on kidney segmentation and they divided the existing kidney segmentation and analysis methods into three categories: 1) threshold and region-based approaches [5, 6]; 2) knowledge-based models [7]; and 3) deformable methods [8, 9]. All the segmentation methods are focused on the automatic kidney segmentation precision. Recall that our goal is to extract the outline of kidneys in these acquisitions for registration. Because of no identified limits, similar gray levels and similar behavior after injection of contrast agent, the kidney is difficult to demarcate from the liver or the spleen [10]. The importance for this segmentation is completeness instead of precision, that is to say, all the kidney component should be extracted in these slices. In order to meet this request, a semi-automatic kidney segmentation framework is proposed. We first segment the kidney from each slice semi-automatically and then reconstruct the kidney volume from the segmented kidney slices. The rigid registration will be performed on these reconstructed volumes.

The similarity measure numerically describes the connection between the two images. Both Collignon et al. [11] and Studholme et al. [12] suggested to use entropy as similarity measure of registration. Once this measure from information theory (entropy) had been introduced for the registration of medical images, another such measure quickly appeared: mutual information (MI). It was first brought forward both by Collignon et al. [13] and by Viola and Wells [14]. Applied to rigid registration, it showed great accuracy and within a few years it became the most investigated measure for medical registration [15]. The method applies the concept of MI to measure the statistical dependence between the image intensities of corresponding voxels in both images, which is assumed to be maximal if the images are geometrically aligned. Other similarity measures such as entropy correlation coefficient [16], correlation ratio [17], Tsallis entropy [18] etc. are also widely used. The choice and the validation of the similarity measures are critical points. Usually the similarity measure is evaluated with the registration result, which involves many other aspects in the registration algorithms (interpolation, optimization method etc.). Skerl et al. [2] proposed an optimization independent protocol to evaluate the similarity measures for rigid registration. This protocol is implemented to choose the most appropriate similarity measure. The similarity measures evaluation for our practical situation will be described in detail in section 2.4.

Besides of the similarity measure, *the optimization method* is also a critical point in the registration method. In our case, there are six parameters: three for translation (*tx*, *ty*, *tz*) and three for rotation (*rx*, *ry*, *rz*) to find. There are many existing optimization methods which can do the multidimensional optimization [19]. There is no “perfect” algorithm for the particular application. We find that the efficient initial parameters set can facilitate the choice of the optimization method. Considering the speed and efficiency, a moment based parameters initialization method is proposed and the downhill simplex method is implemented as an optimization method.

Outline of our work

According to the analysis above, we construct a kidney-centered registration framework: the kidneys are first extracted from the abdominal acquisitions and then the registration is performed between the kidneys instead of the abdomens. The intensity based registration method is chosen. Similarity measure is a critical aspect for the intensity based registration algorithms. We implement an optimization independent protocol proposed by Skerl et al. [2] to choose the suitable similarity measure for our situation. The MI based similarity measure is chosen and some aspects which affect the calculation of MI such as the joint histogram resolution or the interpolation method are discussed. Besides the similarity measure, the optimization method in the registration procedure is also investigated. We propose to use geometric moments to calculate the initial parameters for the optimization method. This initialization makes the choice of the optimization method more unrestricted. Finally, the results on both synthetic data and real data are given.

2.2. Intra subject kidney registration framework

2.2.1. Registration transformations and framework

The transformation for rigid registration only involves translation and rotation so that it is enough to express this kind of transformation by a matrix. The matrix expressed transformation between volumes is illustrated in Fig. 2.1. According to the relationship of transform matrix, the registration framework is described as follows:

Rough semi-automatic extraction of the kidney from the abdominal volume. During extraction, we keep a matrix (denoted respectively **T1** and **T2** for the two sub-volumes) to express the spatial relationship between the kidney volume and the corresponding abdominal volume.

Registration of the two extracted kidneys by maximizing the MI. This registration gives the transform matrix **Tk**.

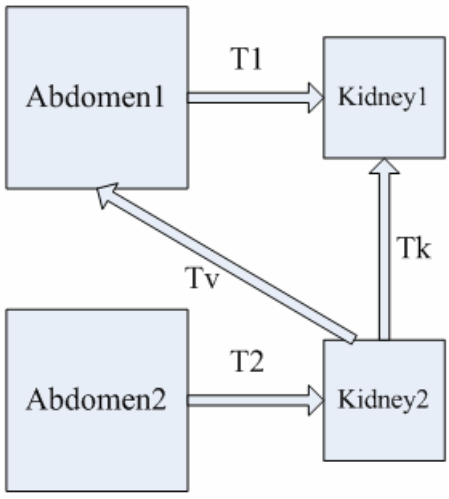


Figure 2.1: Transformation between volumes

In fact, the registration is done between the two sub-volumes (extracted kidney volumes). By keeping the relationship between the extracted volumes and the original

volumes (**T1** and **T2**), we can illustrate the aligned kidney in one of the abdominal volumes (the original acquisitions) by calculating **Tv**:

$$\mathbf{Tv} = \mathbf{Tk} \cdot \mathbf{T1}^{-1}$$

So this registration process is called a kidney-centered registration.

2.2.2. Coordinate systems definition

Each volume that is involved in a registration will be referred to a coordinate system, which defines a space for that volume. When we get the transformation between two kidney volumes, we expect to have a real world expression (millimeter for translation and degree for rotation). If the transformation unit is defined according to the real world coordinate, the result will be more related to the reality. But usually the volumes are described by voxels, which are defined in the image coordinate system. So when we do registration, we cannot calculate the transformation between the kidney volumes directly. In order to do registration between different volumes in the real world space, we define the following coordinate systems.

The first is *volume coordinate system*, which is called *IJK coordinate system*. This coordinate system is used as the original volume description. The directions are defined as follows: I, slice left to right; J, slice top to down; K, slice's orthogonal direction.

The second is *human coordinate system*, which is called *RAS coordinate system*. As shown in Fig. 2.2, the directions are defined as follows: R, human left to right; A, human front to back; S, human bottom to top. The measurements are in millimeter in this coordinate system.

In order to unify all volumes after registration, we defined a coordinate system called *REF coordinate system*. The REF coordinate system still belongs to the range of RAS coordinate system. The difference is that it is the RAS coordinate system of the reference volume. After registration, all volumes are transformed into this coordinate system so that they can be merged.

Finally, the *window coordinate system* is the coordinate system of the screen window.

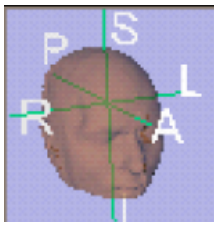


Figure 2.2: RAS coordinate directions definition

Transformation between different coordinate systems is illustrated in Fig. 2.3. **T7** is used to get one cut plane data of the merged volume in order to show it in the screen window. During registration, **T1**, **T2**, **T3** and **T4** will keep changeless because they describe the natural relationships between the coordinates. If one volume, for example volume 1, is considered as *the reference volume*, the transform **T5** will be an identity matrix and the transformation of *the floating volume* **T6** will be calculated according to the registration result of the two kidney volumes. That is to say, if we fixed the volume 1 as the reference volume, **T6** will be equal to the **Tk** mentioned before.

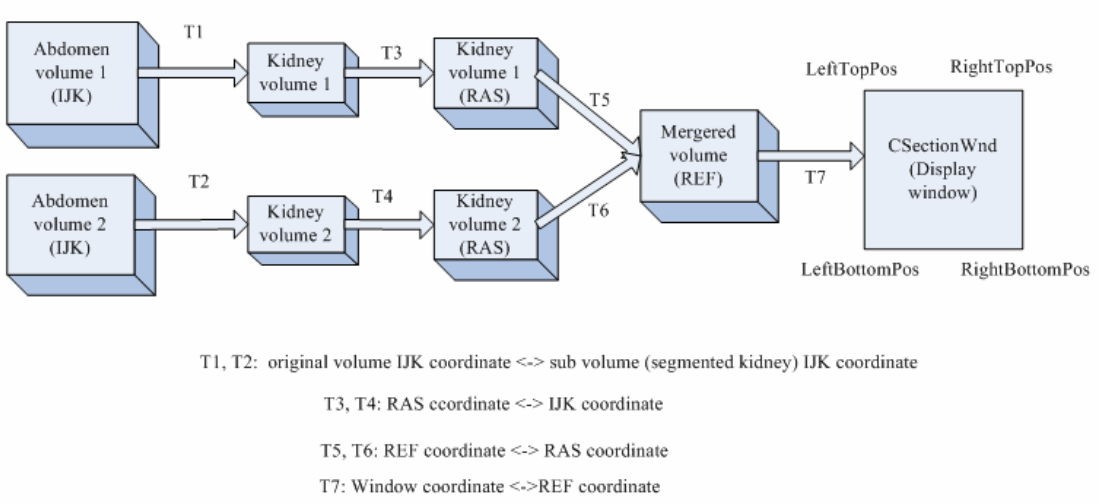


Figure 2.3: Transformation between coordinate systems

This kind of coordinate system definition has the ability to involve more volumes into the registration. If there are more than two kidney volumes to be registered, we can fix one volume as the reference volume and consider the others as floating volumes. The REF coordinate is equal to the RAS coordinate of the reference volume. All the float volumes will be transformed to the REF coordinate after registration so that we can get the transformation between the kidneys through the unified REF coordinate.

2.3. Volume extraction

2.3.1. Extraction method

Automatically extracting the kidney volume from the abdomen volume is difficult and time-consuming. We develop a semi-automatic snake-based segmentation method to broadly extract the kidney volume in a reasonable time. As the input data is a series of CT slices, we first segment the kidney slice by slice and then reconstruct the kidney volume from the segmented kidney slice.

The extraction steps are as follows:

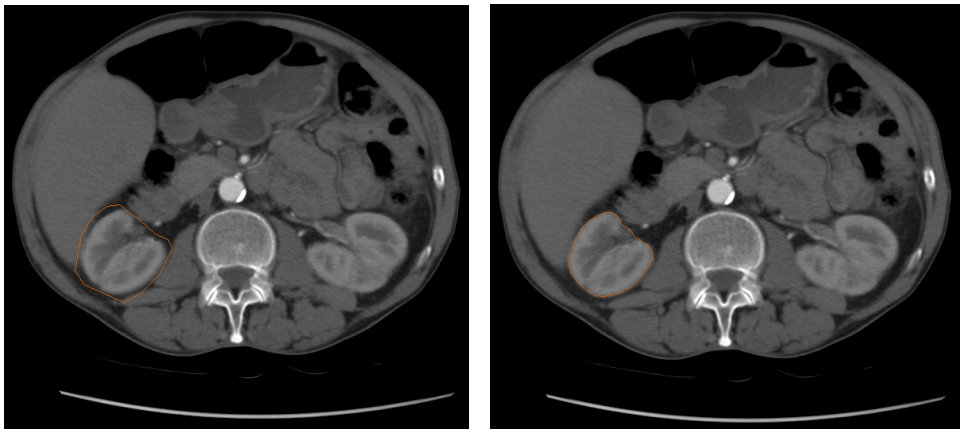
1) Roughly initialize the kidney contour manually by picking some points to form a bounding contour, then use the discrete dynamic contour model [20-22] to segment the external kidney contour in one slice (as shown in Fig. 2.4). The result on one slice is then propagated to the neighboring slice as an initialization. This propagated snake will be automatically adjusted to the new data by Lobregt and Viergever's model. This propagation procedure continues until it reaches the last slice. During this automatic process, sometimes manual corrections need to be performed. In this way, the kidney external contour is extracted on each slice.

2) Connect these extracted kidney contours by a string matching based contour tiling method to form a kidney surface [23].

3) Fill this surface to get a binary volume.

4) Do 3D dilatation to the binary kidney volume to make sure that the kidney information is inside.

5) Intersect the binary volume with the original grey volume.



(a) (b)

Figure 2.4: 2D snake segmentation. (a): Manual initialization; (b): Contour adjustment after applying the discrete dynamic contour model.

After these five steps, the kidney volume is achieved. We can make sure that the kidneys are inside and so that the registration between kidney volumes can be performed. The total amount of time of this semi-automatic extraction for one volume is about three minutes for a non-trained user.

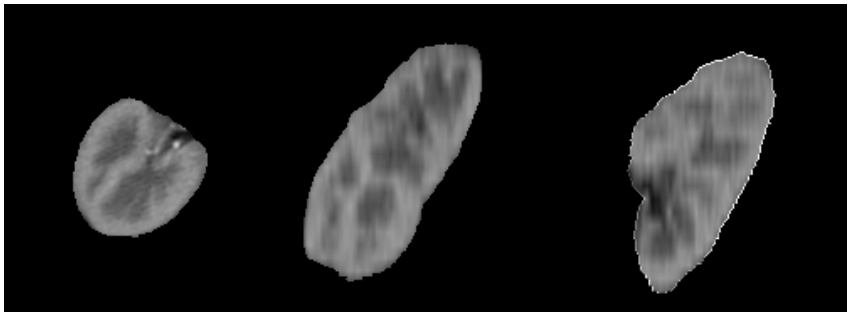
2.3.2. Extraction results and synthetic experimental data

Extraction results

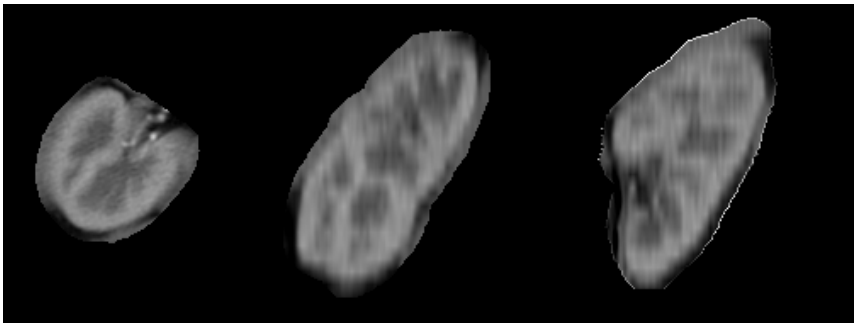
Several experiments have been done for the extraction of the kidney volumes. Fig. 2.5 illustrates one of them. We display three slices of the volume from three axis direction of the RAS coordinate system: axial, coronary and sagittal. Fig. 2.5(a) and (b) show the volume without and with dilatation (step 4 in the extraction process) respectively. We can see that the dilatation operation can make sure that all the kidney information including the kidney boundary is contained within the volume.

Synthetic experimental dataset for the validation of registration methods

In order to validate the registration methods for our special case, we create a synthetic experimental dataset composed by two volumes. Denote one extracted kidney volume as **A**, the second volume is created by applying a transformation matrix to **A** and then get the corresponding kidney volume **B**. Therefore, the transformation between volume **A** and **B** is already known, which can be considered as the registration "*golden standard*". As our real registration is between monomodal images, the synthetic data is similar to the real case. For the following sections, we do the method validation on these kidney volume pairs where the transformation between them is already known.



(a)



(b)

Figure 2.5: Kidney volume without and with dilatation. The slices from left to right are axial, coronary and sagittal images. (a): without dilatation; (b): with dilatation.

2.4. Evaluation of the registration metrics

2.4.1. Similarity measures

We have implemented and evaluated five similarity measures: 1) mutual information [16, 24]; 2) entropy correlation coefficient [16]; 3) normalized mutual information [12]; 4) correlation ratio [17]; 5) Tsallis entropy [18]. All similarity measures were applied to overlapping voxels of floating (transformed) and reference (target) volumes and formulated on the 2D joint histogram or joint probability distribution of the intensities of the two volumes.

Consider two digital images, *X* and *Y*, with *Lx* and *Ly* possible gray levels, respectively. We may assume that both images have *N* pixels over a common spatial domain $\Omega = \Omega_1 \cap \Omega_2$, where Ω_1 and Ω_2 are the spatial domain of *X* and *Y*, respectively. Treating each pixel intensity value as a symbol of an underlying discrete random value, and an image as a long sequence of symbol observations of the discrete random value, we have the following probability descriptions:

$$p_{XY}(i, j) = \frac{h_{XY}(i, j)}{\sum_{i,j} h_{XY}(i, j)} \tag{2.1}$$

Where $h_{XY}(i, j)$ is the number of pixel pairs in (*X*, *Y*) having intensity combination (*i*, *j*). Note that $h_{XY}(i, j)$ is their joint histogram, evaluated over Ω. Eq. (2.1) indicates that dividing the entries by the total number of entries in the histogram yields a probability distribution.

The marginal probability distributions can be found by summing over the rows and columns, respectively, of the joint probability:

$$p_X(i) = \sum_j p_{XY}(i, j) \tag{2.2}$$

$$p_Y(j) = \sum_i p_{XY}(i, j) \tag{2.3}$$

Let $H(X)$ and $H(Y)$ be the Shannon entropies of images X and Y, respectively, and let $H(X, Y)$ be the joint entropy of the two images, in which:

$$H(\cdot) = -\sum p(\cdot)\log(p(\cdot)) \tag{2.4}$$

The mathematical definitions of the five implemented similarity measures are the following:

1) mutual information [16, 24]

$$MI(X,Y) = H(X) + H(Y) - H(X,Y) \tag{2.5}$$

2) entropy correlation coefficient [16]

$$ECC(X,Y) = \frac{2MI(X,Y)}{H(X) + H(Y)} \tag{2.6}$$

3) normalized mutual information [12]

$$NMI(X,Y) = \frac{H(X) + H(Y)}{H(X,Y)} \tag{2.7}$$

4) correlation ratio [17]

$$Correlation(Y \mid X) = \frac{Var[E(Y \mid X)]}{Var(X)} \tag{2.8}$$

where E[.] and *Var*[.] denote the expectation and the variance respectively.

5) Tsallis mutual information [18]

$$MI_\alpha(X,Y) = H_\alpha(X) + H_\alpha(Y) - (1-\alpha)H_\alpha(X)H_\alpha(Y) - H_\alpha(X,Y) \tag{2.9}$$

where $H_\alpha(x) = (1-\alpha)^{-1}(\sum_x p(x)^\alpha - 1)$ is the Tsallis entropy.

2.4.2. Evaluation protocol

The rigid registration similarity measures can be evaluated by an optimization independent protocol proposed by Skerl et al. [2]. This protocol requires that the

"golden standard" registration transform between the images is available. The main idea of this protocol is to first sample some positions centered by the "golden standard" position in the normalized parameter space; then calculate the similarity measures on these positions; finally define some statistical estimates of a similarity measure. Therefore, the evaluation protocol is composed by the following three steps:

1. A normalized parameter space is defined so that a unit change in any transformation parameter causes the same mean shift in voxels. It is then sampled over a hyper-sphere with radius R centered on the gold standard parameters into N rays, each of them composed of M+1 equally spaced pose parameters. If there are no gold standard parameters in our testing data, instead we will use the approximate parameters which are gotten by our registration method.

2. The similarity measure is computed for all NM+1 samples.

3. Five distinct statistical estimates are computed to characterize similarity measures in the neighborhood of the expected maximum: accuracy, measure of distinctiveness, capture range, risk of non-convergence, number of local minima. The definitions of these measures are as following:

   - Accuracy (ACC) is the root mean-square distances between the hypersphere origin and the global maxima over each line.

   - Measure of distinctiveness ($DO(r)$) is the average change of similarity measures near the global maximum and is defined as a function of distance $r = k \cdot \delta$ from the maximum, where $\delta = 2R/M$ is the distance between two consecutive points along a line and k is the number of steps from the origin.

   - Capture range (CR) is the smallest of the N distances between positions of global maxima and closest minima along each line.

   - Number of local maxima ($NOM(r)$) is the total maxima number of distance r from each of the N global maxima.

   - Risk of non-convergence ($RON(r)$) describes the behavior of a similarity measure around the N global maxima. It is defined as the average of positive gradients within distance r from each of the N global maxima.

2.4.3. Evaluation results and discussions

2.4.3.1. Results on synthetic data

In order to evaluate the metrics, we formed a synthetic kidney volume by translating and rotating a real volume by known parameters (tx_T, ty_T, tz_T, rx_T, ry_T, rz_T). This synthetic volume is considered as a floating volume during the registration metric evaluation. The information of the kidney volume is listed in Table 2.1 and the parameters we randomly choose are listed in Table 2.2. These known parameters are considered as the gold standard parameters of the evaluation method proposed by Skerl et al. [2].

Table 2.1: Testing image sizes, voxel sizes, translation and rotation units of normalized parametrical space, radius *R*, number of lines *N*, number of points along a line *M*, and distance δ between two consecutive points along a line for the image set.

Image size (voxels)			Voxel size (mm)			Unit (mm)	Unit (°)	*R* (mm)	*N*	*M*	δ (mm)
X	*Y*	*Z*	*X*	*Y*	*Z*						
124	114	178	0.65	0.65	0.65	7.41	5.3285	22	50	200	0.22

Table 2.2: Parameters used for constructing the synthetic kidney volume, being used as the golden standard

tx_T (mm)	ty_T (mm)	tz_T (mm)	rx_T (°)	ry_T (°)	rz_T (°)
-3.01	-9.73	-3.23	5.97	-4.63	2.71

Table 2.3: Accuracy (*ACC*), distinctiveness of optimum (*DO*), capture range (*CR*), number of local maxima (*NOM*), and risk of nonconvergence (*RON*) of five similarity measures applied to the image mentioned above. The numbers printed in bold represent the best and the ones in italic the worst values in a column.

	ACC	*DO*(*R*)	*CR*	*NOM*(*R*)	*RON*(*R*)
MI	**0**	**0.006160**	26.083166	**0**	364.58
ECC	**0**	0.005761	26.083166	**0**	452.85
NMI	**0**	0.005306	26.083166	**0**	**113.23**
Correlation	**0**	0.005774	**27.713364**	**0**	*663.23*
Tsallist	**0**	*0.004444*	*21.192572*	**0**	274.19

The evaluation results are listed in Table 2.3. From the result, we can see that there is no significant difference between these registration metrics according to these evaluation measures, but the MI based metrics (MI and NMI) express a small advantage.

2.4.3.2. Results on real data

In the real situation, we have no gold standard parameters because the real transformation between the two kidney volumes is unknown. Approximately, we use the parameters we get through MI based registration method as the gold standard parameters, as shown in Table 2.4. Perhaps the approximated parameters are not exactly at the correct position, but at least they can be close to the global maxima and the evaluation of the measure of distinctiveness (*DO*), the number of local maxima (*NOM*) and the risk of nonconvergence (*RON*) is less affected by the position of the global maxima. So for the real situation, we can also evaluate the registration metric by calculating these measures while assuming the approximated gold standard parameters.

Table 2.4: Parameters after registration to be used as the golden standard

X (mm)	*Y* (mm)	*Z* (mm)	*rx* (°)	*ry* (°)	*rz* (°)
2.79	-4.105	-1.403	1.364	1.179	-1.446

The evaluation results are listed in Table 2.5. The experimental results on the real data accord with the experiments on synthetic data. The MI based metrics (MI and NMI) appear a small advantage comparing to the others.

Table 2.5: Distinctiveness of optimum (*DO*), capture range (*CR*), number of local maxima (*NOM*), and risk of nonconvergence (*RON*) of five similarity measures applied to the image mentioned above. The numbers printed in bold represent the best and the ones in italic the worst values in a column.

	DO(*R*)	*CR*	*NOM*(*R*)	*RON*(*R*)
MI	**0.006139**	**26.083168**	**0**	509.01
ECC	0.006133	**26.083168**	**0**	*588.60*
NMI	0.005268	**26.083168**	**0**	**137.99**
Correlation	0.005014	**26.083168**	**0**	558.48
Tsallist	*0.004358*	*21.192574*	**0**	303.29

2.4.3.3. Discussions and conclusions

The experiments compare different registration metrics for our practical situation. From the results, we cannot definitely say that which one is the best because of their close evaluation properties. But the MI based metrics (MI and NMI) still show the advantages during the experiment. So we can reach the conclusion that the MI based metrics are suitable in our situation.

2.5. Optimization method

As MI is used as the matching metric, registration can be performed by optimizing this similarity criterion. There are many existing optimization methods which can do the multidimensional optimization [19]: *downhill simplex method*, *Powell's method* (one of the *direction-set methods*) and *simulated annealing methods* etc. Downhill simplex method and Powell's method are relatively fast, but they have the problem to drop into local extrema; simulated annealing methods address directly the problem of finding global extrema in the presence of large numbers of undesired local extrema, but the computation is time-consuming. There is no "perfect" algorithm for the particular application. They are two critical points in all the optimization methods: the choice of the initial parameters set (tx_0, ty_0, tz_0, rx_0, ry_0, rz_0) and the choice of the optimization method itself. We will make our choice by analyzing the two issues in this section.

2.5.1. Initial parameters

In order to analyze the effect of the initial parameters set (tx_0, ty_0, tz_0, rx_0, ry_0, rz_0), we displayed the MI variation within the parametric searching space in order to estimate the presence or not of local extrema.

For this, we formed a synthetic kidney volume by translating and rotating a real volume by known parameters (tx_T, ty_T, tz_T, rx_T, ry_T, rz_T). From the initial parameters set (tx_0, ty_0, tz_0, rx_0, ry_0, rz_0), we sampled the parametric search space and measure the MI for each (tx, ty, tz, rx, ry, rz).

In order to present some graspable results, we fixed constant the value of $ty=ty_0$, $tz=tz_0$, $ry=ry_0$, $rz=rz_0$. Only tx and rx are varying and the MI variation can be seen as a surface as shown in Fig 2.6.

Fig 2.6(a) illustrates the MI surface in the situation that the parameters set is initialized by the real value: $ty_0=ty_T$, $tz_0=tz_T$, $ry_0=ry_T$, $rz_0=rz_T$. We can see that in this situation, the MI surface is smooth and the global extremum is obvious. The extremum can be achieved by any optimization method easily. Fig 2.6(b) sets $ty_0=0$,

tz_0=0, ry_0=0, rz_0=0 and it can be seen that many local extrema appear and the global extremum is inconspicuous. Fig 2.6(c) is the situation that ty_0=ty_T=0, tz_0=tz_T=0, ry_0=ry_T=0, rz_0=rz_T=0. We can see that although there are still many local extrema, the global extremum is obvious and easy to achieve.

From the analysis above we can get the conclusion that the more initial parameters close to their real values the smoother MI surface will be. According to the experiment result, we utilize the characteristics of the image geometric moments [25] to initialize the parameters instead of initialization by zero or random values.

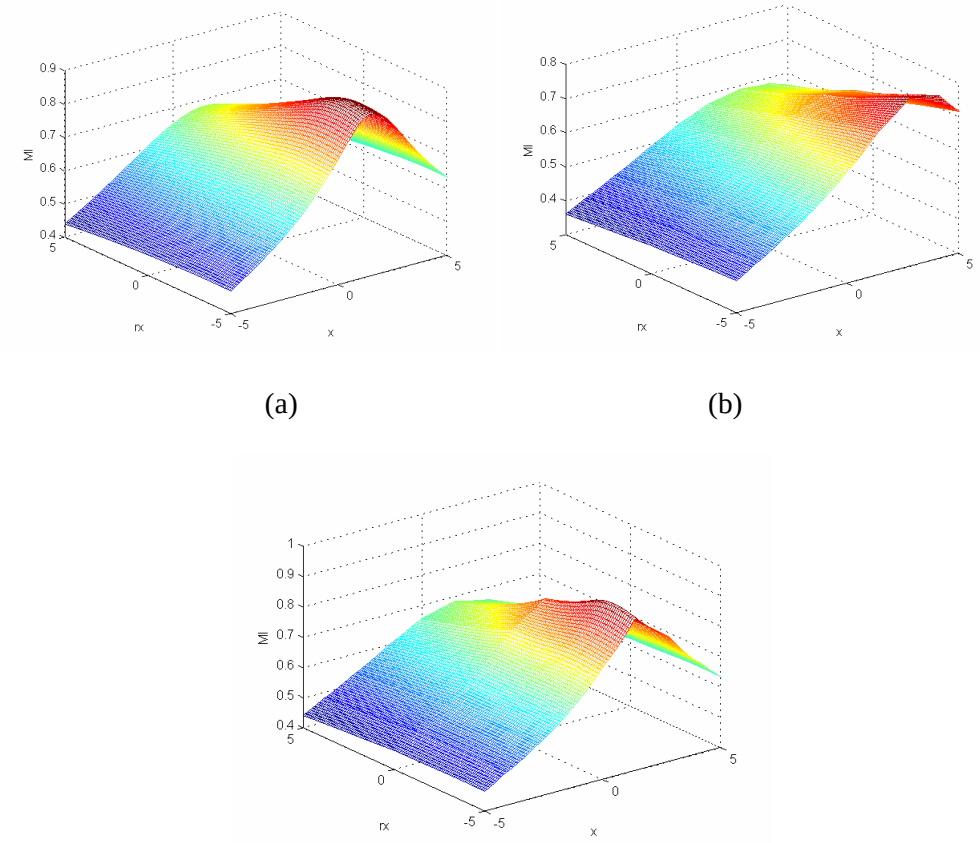


Figure 2.6: *tx*-*rx* searching space MI surface for different parameter initializations. (a): *ty*, *ry*, *tz*, *rz* are initialized on the real values when the real values are not zero; (b): *ty*, *ry*, *tz*, *rz* are initialized to zero when the real values are not zero; (c): *ty*, *ry*, *tz*, *rz* are all initialized to zero when the real values are zero;

For a three dimensional discrete image $f(i,j,k)$, its geometric moments of order $u+v+w$ are defined as:

$$m_{uvw} = \sum_i \sum_j \sum_k f(i, j, k) i^u j^v k^w \tag{2.10}$$

The first order geometric moments denotes the volume center of gravity. The translation between the centers of gravity can be the initial value of (tx_0, ty_0, tz_0). The second order moments can determine the 2 volumes main direction axes. A rotation

matrix can be estimated from these main axes. Then the three initial parameters (rx_0, ry_0, rz_0) can be achieved from the matrix.

2.5.2. Optimization method

The registration process is a multi-variable optimization problem. As mentioned before, there exist many optimization methods. Among these methods, the downhill simplex method is one the most used. Although compared to the simulated annealing methods it has more probability to meet local extrema, simplex method is accurate enough after parameters initialization and it is faster than the other methods.

2.6. Implementation details

2.6.1. Interpolation and outside point processing

Denote the floating volume by **F** and the reference volume by **R**, when transforming a point *s* from **F** to **R** by the transformation **Tk**, usually the resulting position *Ts* is not exactly on the grid of **R** (as illustrated in Fig. 2.7), so that interpolation is required to estimate the grey value of the resulting point from the neighborhood n_i.

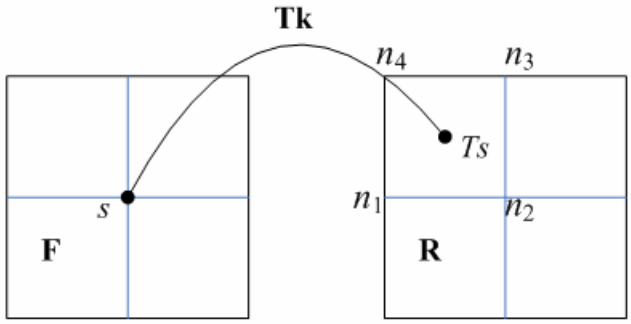


Figure 2.7: The illustration of the mapping from floating volume **F** to reference volume **R**. With the transformation **Tk**, the grid position *s* in **F** is transformed to *Ts* in **R.**

For calculating the mutual information between the two volumes, the joint histogram *h*(**F**,**R**) is necessary. In order to calculate the joint histogram with the transformation **Tk**, an interpolation method is required. We studied three interpolation methods to calculate the joint histogram; these methods are illustrated (in 2D) in Fig. 2.8:

– Nearest neighbor (NN) interpolation of **R** get the nearest grid point value for the position *Ts*. For example, if n_3 is the closest grid point, the calculation formula for the corresponding intensity pair related to the current sample position (*s*) is:

$$\begin{aligned} &\arg\min_{n_i} d(Ts, n_i) = n_3 \\ &f_r(Ts) = f_r(n_3) \\ &h(f(s), f_r(Ts)) += 1 \end{aligned} \tag{2.11}$$

where $f(s)$ denote the image intensity in the floating image **F** at position s, $f_r(Ts)$ denote the intensity at the transformed position (Ts) in the reference image **R** and $h(f(s), f_r(Ts))$ denotes the joint histogram bin for the intensity pair $(f(s), f_r(Ts))$. We can see that NN interpolation is generally insufficient to guarantee subvoxel accuracy, as it is insensitive to translations up to one voxel.

– Trilinear (TRI) interpolation is more reasonable with the calculating formula:

$$f_r(Ts) = \sum_i w_i \cdot f_r(n_i)$$
$$h(f(s), f_r(Ts)) += 1 \qquad (2.12)$$

With the constraint that: $\sum_i w_i(Ts) = 1$. But we can see that this method will introduce new intensity values which are originally not present in the reference image.

– In order to avoid this problem, Maes et al. [16] proposed trilinear partial volume distribution (PV) interpolation to update the joint histogram for each voxel pair. The joint corresponding joint histogram calculating formula is:

$$\forall i : h(f(s), f_r(n_i)) += w_i \qquad (2.13)$$

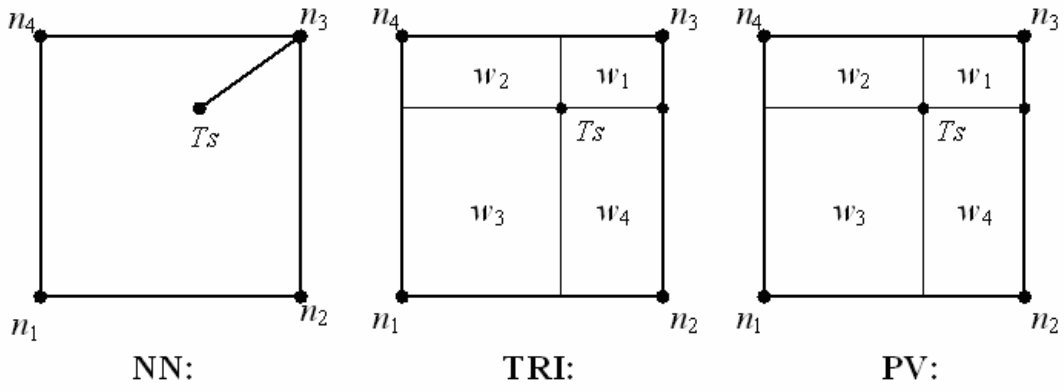


Figure 2.8: Graphical illustration of NN, TRI and PV interpolation in 2-D

NN and TRI interpolation find the reference image intensity value at position Ts and update the corresponding joint histogram entry, while PV interpolation distributes the contribution of this sample over multiple histogram entries defined by its NN intensities, using the same weights as for TRI interpolation.

If the resulting position Ts is outside the spatial range of **R**, we use an image background expansion method, which is to look for the nearest position in **R** and use the corresponding intensity as the current point intensity.

2.6.2. Histogram resolution

The intensity values of the original medical images have a very large range up to 4096. This huge range induces a very large joint histogram. The computation time of MI is directly related to the size of the joint histogram. This computation can be speeded up if we define larger bins to the histogram which can be obtained by

rescaling the image intensity values before registration. Linear mapping is a practical method. The formula to rescale an intensity value r from image range $[r_1, r_2]$ to $[s_1, s_2]$ by linear mapping is as follows:

$$s = \begin{cases} \dfrac{s_2 - s_1}{r_2 - r_1}(r - r_1) + s_1, & r_1 \le r \le r_2 \\ s_1, & r < r_1 \\ s_2, & r > r_2 \end{cases} \tag{2.14}$$

Lin et al. [26] discussed the affection of number of levels on mutual information based medical image registration. The authors indicated that the value of mutual information is reduced when the number of levels in both images is compressed. We proposed an experiment which highlighted this belief, as shown in Fig. 2.9.

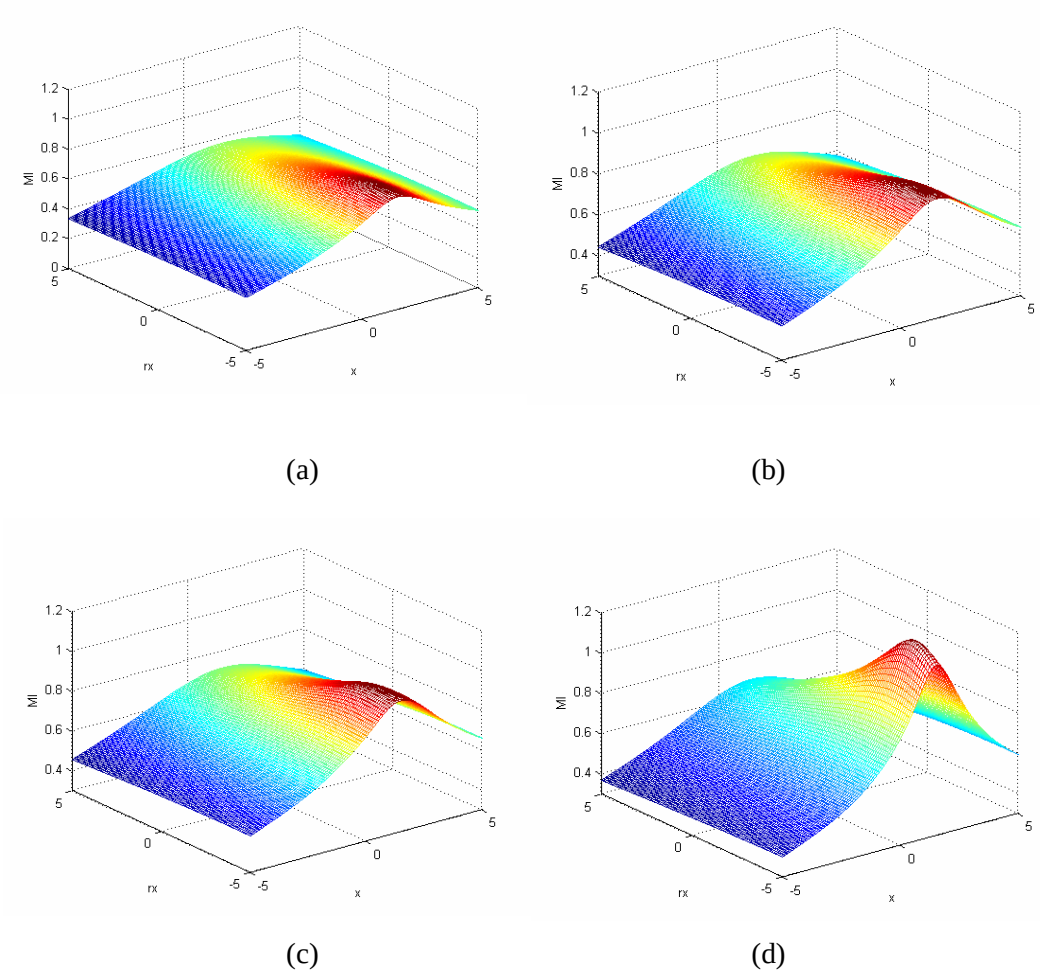


Figure 2.9: MI surfaces calculated from the real volume and the synthetic volume with different number of levels. The parameters to get float volume are: *tx*=1.46, *ty*= -1.01, *tz*=1.9, *rx*= -3.835, *ry*= -3.295, *rz*=2.95. The surface was gotten when *ty*, *ry*, *tz*, *rz* all equal to the original parameters, *tx* and *rx* ranged from -5 to 5. (a) number of levels = 16; (b) number of levels = 64; (c) number of levels = 128; (d) number of levels = 256.

It can be seen that with the growth of number of levels, the MI values become higher and the MI surface becomes sharper. But the high number of levels will cause a great calculation burden during the iterative registration process because the joint histogram will be recalculated on each iteration step. Lin et al. [26] reached the conclusion that rescaling the intensity values into [0, 63] is an excellent tradeoff between the accuracy and the computation time cost. We implemented this result in our registration process.

2.7. Experimenal results

The results on synthetic and real data are presented in this section.

2.7.1. Experimental data

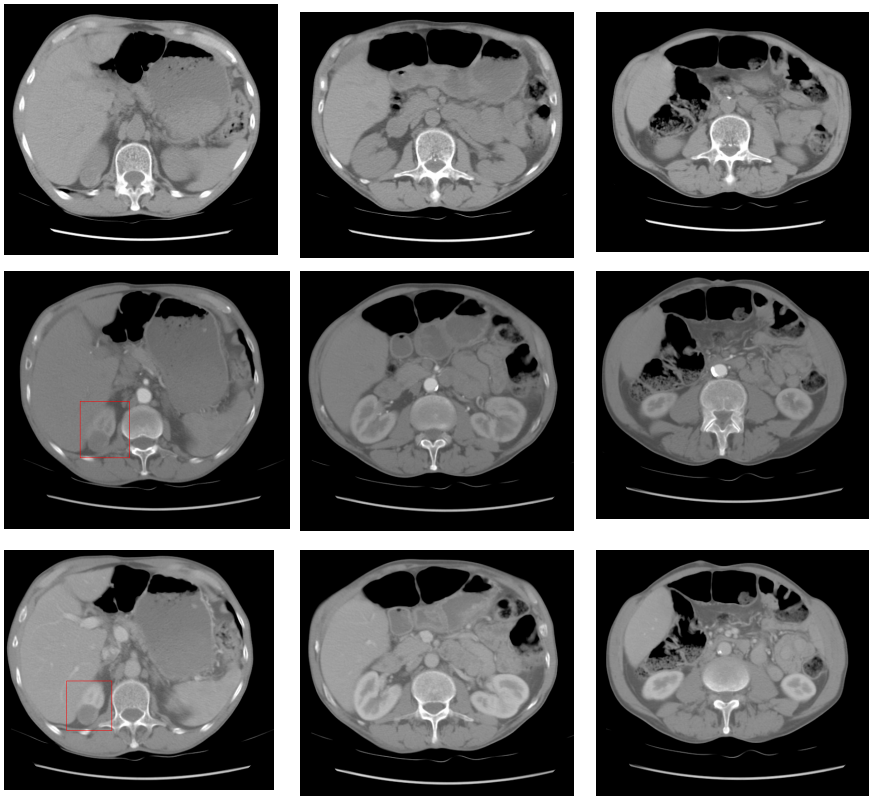


Figure 2.10: Some of the slices. From up to down: (a) slices without injection; (b) slices with arterial system enhanced; (c) slices with venous system enhanced.

We take a series of acquisitions taken by the GE product CTA1.0CO for the real testing. Some of the slices are illustrated in Fig. 2.10. The pixel resolution is 0.65039mm. The slice thickness and interslice spacing are 5mm. The first acquisition is realized without injection of contrast agent and informs the surgeon about intern morphology of the patient. The second one, taken just after a contrast medium injection, reveals the renal arterial system. Obtained just a time later, the third acquisition enhances the venous and renal parenchyma vascularization. These two acquisitions give also information about the nature and the location of the renal carcinoma (marked by the red square in the first slice of Fig. 2.10 (b) and (c)).

After extracting the kidney volume form the abdomen acquisitions, we form the synthetic data by transforming the extracted kidney volume with the randomly set transform parameters, as introduced in section 2.3.2.

2.7.2. Experimental results on synthetic data

To evaluate the registration process, an experiment is done on synthetic data. As introduced before, a synthetic volume is build by translating and rotating the real kidney volume. On each experiment, we set a group of translation and rotation parameters to build up a synthetic volume and then do registration between the original volume and the synthetic one. More that thirty experiments have been performed with randomly chosen parameters and the errors between the estimated parameters and the real ones are estimated, some of the results are listed in Table 2.6. From all the results, the maximal translation error is less than 0.08 mm and the maximum rotation angle error is less than 0.1°.

Table 2.6: Registration result between the real kidney volume and the synthetic kidney volume: In each group, the first line is the transformation parameters used to build the synthetic volume and the second line is the registration result.

Group ID	*x*(*mm*)	*y*(*mm*)	*rz*(*mm*)	*rx*(°)	*ry*(°)	*rz*(°)
1	-6.75	5.18	-5.11	-2.26	-5.99	-5.29
	-6.7418420	5.2116291	-5.1120056	-2.2842028	-5.9603878	-5.3083628
2	-4.74	-6.08	-7.58	7.55	1.02	-1.83
	-4.7602659	-6.0350842	-7.4883824	7.5110615	1.0417165	-1.8512405
3	-1.36	2.14	-2.73	6.01	-3.07	-4.67
	-1.3221311	2.1003808	-2.8529953	6.0986129	-3.0535301	-4.6607840
4	6.95	-9.32	4.75	-3.15	-2.68	7.44
	6.9570326	-9.2632615	4.8530425	-3.2897572	-2.6222272	7.4076053
5	1.3	-3.22	2.0	-4.46	8.9	-0.60
	1.3553261	-3.2410673	1.9061267	-4.4989978	8.9839812	-0.5952921
6	1.53	3.19	4.07	3.09	-2.34	-0.04
	1.5547357	3.2102461	4.0694638	3.0934631	-2.3287156	-0.0547939
7	9.06	0.43	7.09	-9.36	-9.10	-2.76
	9.0848879	0.4531154	7.1147671	-9.4261618	-9.0525263	-2.7799144
8	8.01	4.76	9.41	4.98	7.49	7.05
	8.0674316	4.7399294	9.2888299	5.0535454	7.5581793	7.0771735
9	-3.94	8.08	-7.55	-4.45	6.41	-8.55
	-3.9445847	8.1179366	-7.5172744	-4.4732718	6.4294913	-8.5622459
10	1.26	-1.75	-9.60	-2.0	-6.79	-5.78
	1.3334176	-1.8234414	-9.8614753	-1.9451294	-6.8072046	-5.7731429

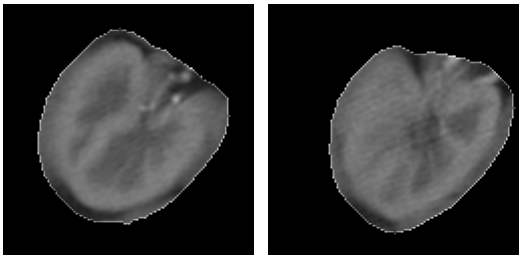
2.7.3. Experimental results on real data

The experimental results are illustrated in both kidney-only form and the whole abdomen form.

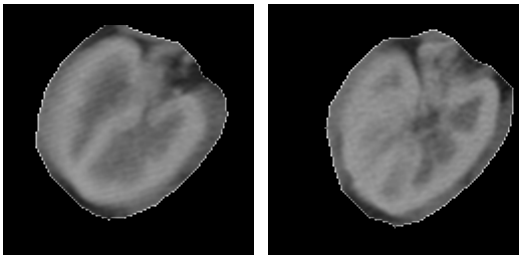
Registration result between extracted kidney volumes

The registration result on the extracted kidney volumes is illustrated in Fig. 2.11. Here we take the registration of the arterial and the venous phase acquisition for example. The extracted kidney volume from the arterial acquisition is considered as the reference volume and the other from the venous acquisition is considered as the floating volume. The first column shows two cut-planes from the reference volume. The second column is the cut-plane taken from the same position of the floating volume before registration. We can see that because of the transformation, the cut-plane images are not fitted to the reference images. The third column is the cut-plane from the floating volume after registration. They are more fitted to the reference cut-

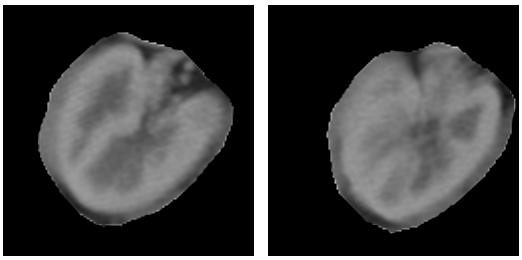
planes now. After the kidney volumes are extracted from the abdomen acquisitions, they are expressed by a 3D volume with similar size. This extraction step implies a rough center alignment so that from the registration result, we can only sense the rotation phenomena without the translation information. When we illustrate the registration result in the abdomen volume, the translation effect is clearer.



(a) two slices from reference volume



(b) two slices from floating volume before registration

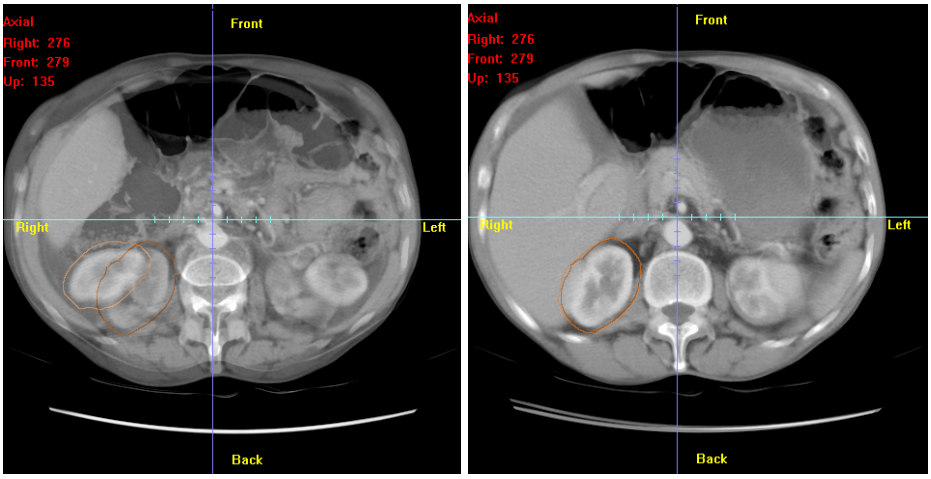


(c) two slices from floating volume after registration

Figure 2.11: Two cut-planes from the registration result of kidney volumes. From top to down: two slices from reference kidney volume, floating kidney volume before registration, floating kidney volume after registration.

Kidney-centered registration result

Two axial slices taken from the same position of the two volumes are merged in Fig 2.12. The bounding lines are the mapping boundaries of the extracted kidney volume. We can see that after registration, although the two volumes are not fitted exactly, the kidneys seem to be well register together.



(a) before registration (b) after registration

Figure 2.12: Kidney-centered registration result.

Using the same registration framework, the kidneys present on all the 3 or 4 CT uroscans can be merged within one same patient-specific referential coordinate.

2.8. Conclusions

In this chapter we present a kidney-centered registration method by local MI maximization. Kidney volumes are extracted and the registration is performed between the extracted kidneys instead of the whole volumes. The registration metrics are evaluated by an optimization independent protocol proposed by Skerl et al. [2] and according to the evaluation result, we choose the MI based metric for our practical situation. As to the optimization method, we focus on the initial parameters setting problem. The importance of the parameters initialization is analyzed and the parameters are initialized by a geometric moments based registration scheme. Some implemental details, such as the choice of number of levels and interpolation method are also discussed.

Experiments are performed on both synthetic and real data. The experimental results demonstrate the effectiveness of the registration method.

References

[1] J. B. Maintz and M. A. Viergever, "A survey of medical image registration," Med Image Anal, vol. 2, pp. 1-36, 1998.

[2] D. Skerl, B. Likar, and F. Pernus, "A protocol for evaluation of similarity measures for rigid registration," IEEE Trans Med Imaging, vol. 25, pp. 779-91, 2006.

[3] J. Fitzpatrick, D. L. G. Hill, and C. J. Maurer, "Image registration," in Handbook of Medical Imaging, vol. 2, Medical Image Processing and Analysis. Bellingham, WA, 2000.

[4] D. T. Lin, C. C. Lei, and S. W. Hung, "Computer-aided kidney segmentation on abdominal CT images," IEEE Trans Inf Technol Biomed, vol. 10, pp. 59-65, 2006.

[5] G. J. Sivewright and P. J. Elliott, "Interactive region and volume growing for segmenting volumes in MR and CT images," Med Inform (Lond), vol. 19, pp. 71-80, 1994.

[6] A. M. Ali, A. A. Farag, and A. S. Ell-Baz, "Graph cuts framework for kidney segmentation with prior shape constraints," presented at MICCAI'07, Brisbane, Australia, pp. 384-92, 2007.

[7] M. Kobashi and L. G. Shapiro, "Knowledge-based organ identification from CT images," Pattern Recogn., vol. 28, pp. 475-491, 1995.

[8] X. Wang, L. He, and W. Wee, "Deformable contour method: A constrained optimization approach," International Journal of Computer Vision, vol. 59, pp. 87-108, 2004.

[9] B. Tsagaan, A. Shimizu, H. Kobatake, and K. Miyakawa, "An automated segmentation method of kidney using statistical information," presented at MICCAI '02, Tokyo, Japan, pp. 556-563, 2002.

[10] J. L. Dillenseger, H. Guillaume, and J. J. Patard, "Spherical harmonics based intrasubject 3-D kidney modeling/registration technique applied on partial information," IEEE Trans Biomed Eng, vol. 53, pp. 2185-93, 2006.

[11] A. Collignon, D. Vandermeulen, P. Suetens, and G. Marchal, "3D multi-modality medical image registration using feature space clustering," presented at CVRMed '95, Nice, France, pp. 195-204, 1995.

[12] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "Multiresolution voxel similarity measures for MR-PET registration," presented at IPMI'95, Ile de Berder, France, pp. 287-2981995.

[13] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, "Automated multi-modality image registration based on information theory," presented at IPMI '95, Ile de Berder, France, pp. 263-274, 1995.

[14] P. Viola and W. M. Wells, "Alignment by maximization of mutual information," presented at ICCV'95, Cambridge, MA, USA, pp. 16-23, 1995.

[15] J. P. Pluim, J. B. Maintz, and M. A. Viergever, "Mutual information based registration of medical images: a survey," IEEE Trans Med Imaging, vol. 22, pp. 986-1004, 2003.

[16] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," IEEE Trans Med Imaging, vol. 16, pp. 187-98, 1997.

[17] A. Roche, G. Malandain, X. Pennec, and N. Ayache, "The Correlation Ratio as a New Similarity Measure for Multimodal Image Registration," presented at MICCAI '98, Cambridge MA, USA, pp. 1115-1124, 1998.

[18] S. Martin, G. Morison, W. Nailon, and T. Durrani, "Fast and accurate image registration using Tsallis entropy and simutaneous perturbation stochastic approximation," Electronics letters, vol. 40, pp. 595-597, 2004.

[19] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes in C, 2nd ed. Cambridge: Cambridge University Press, 1992.

[20] F. Leitner and P. Cinquin, "Complex topology 3-D objects segmentation," presented at proc. SPIE - Model-Based Vision Development and Tools Boston MA, pp. 13-26, 1991.

[21] F. Leitner and P. Cinquin, "Dynamic Segmentation : Detecting Complex Topology 3D-object," presented at Conf of IEEE EMBS (EMBS'91), Orlando, pp. 295-296, 1991.

[22] S. Lobregt and M. A. Viergever, "A discrete dynamic contour model," IEEE Trans Med Imaging, vol. 14, pp. 12-24, 1995.

[23] M. Chen, "Research on surface reconstruction," in Department of Biology Science and Medical Engineering, vol. Master. NanJing: Southeast University, 2005 (Chinese).

[24] W. M. Wells, W. L. Grimson, R. Kikinis, and F. A. Jolesz, "Adaptive segmentation of MRI data," IEEE Trans Med Imaging, vol. 15, pp. 429-42, 1996.

[25] T. L. Faber and E. M. Stokely, "Orientation of 3-D structures in medical images," IEEE Trans Pat Anal and Mach Intel vol. 10, pp. 626-633, 1988.

[26] J. Lin, Z. Gao, B. Xu, Y. Cao, and Z. Yingjian, "The affection of grey levels on mutual information based medical image registration," presented at Conf. of IEEE EMBS (EMBC'03), Cancun, Mexico, pp. 1747-50, 2004.

Chapter 3: Vectorial volume statistical classification

The registered datasets contain full anatomical information about the same patient. To visualize the vectorial datasets, one of the most important parts is to differentiate the objects in the vectorial datasets, which is called classification or segmentation.

In the practical situation, the datasets, especially those generated by scanning modalities, contain a complex combination of values. One of the main reasons for this complexity is the partial volume effect (PVE). It is the effect wherein insufficient image resolution leads to a mixing of different tissue types (materials) within a voxel. So that, during classification, a more reasonable way is to give out the probability distributions of different material types for each voxel instead of a definite material label.

Gaussian mixture model is often used in probability classification problem, because of its ability to classify both scalar and vectorial datasets. But it relies only on the intensity distributions, which will lead a misclassification on the inhomogeneous regions with noise and the partial volume boundaries. G. Kindlmann noted this problem in his tutorial [1] that "histograms/scatter-plots entirely loose spatial information" and he asked if there would be "any way to keep some of it?". In order to integrate the spatial information into classification procedure, a neighborhood weighted Gaussian mixture model is proposed in this section. Expectation-maximization (EM) algorithm is used as optimization method. The experiments demonstrate that the proposed method can get a better classification result and is less affected by the noise.

3.1. Introduction

After this registration process discussed in chapter 2, we will get a multi-volume dataset which can form a vectorial volume. A combined presentation of registered volume datasets will give a better illustration of the anatomical structures. Suitable methods should be found to show the anatomical structures inside the vectorial volume.

For the visualization of the spatial aligned multi-volume, most of the existing methods [2-5] differentiate the objects from the individual volumes and then combine them together during the visualization procedure. Roettger et al. [4] proposed to add spatial information to multi-dimensional transfer functions in order to separate as many features as possible in the scalar volume and the authors also mentioned its ability to deal with multi-volume dataset. Firle and Keil [3] did further research on this spatial transfer function based method and they focused on the multi-volume situation. Wilson et al. [5] applied different rendering styles for different volumes in order to visualize the multi-volume. But actually, these volumes are different acquisitions taken from the same patient so that they are not independent. While analyzing the registered volume dataset, each sample point should contain several elements which are sampled from the corresponding volumes. So that we can form a *vectorial volume* dataset, in which each voxel contains a vector of n elements corresponding to the information of the CT uroscan acquisitions (n is equal to the number of acquisitions, three to four in our practical case). In order to get the material (tissue) distribution information of this vectorial volume, a multi-dimensional *classification* (*segmentation*) method should be performed.

Due to *partial volume effects* (PVE), the object boundaries are usually combined by several materials. Getting the material probabilities instead of assigning a definite material to the voxels (especially the boundary voxels) will be more conformable to the reality. Opposite to assign definite labels to voxels which is called *hard segmentation*, segmentations that allow regions or classes to overlap are called *soft segmentation* [6]. Because of partial volume effects, soft segmentations are important in medical image segmentation. Based on the analysis of the data to be segmented, we are looking for a soft segmentation method for vectorial volume.

In the range of segmentation methods, *clustering* algorithms are termed *unsupervised* classification methods which organize unlabeled feature vectors into clusters or "natural groups" such that samples within a cluster are more similar to each other than samples belonging to different clusters. The three most commonly used clustering methods are the *K*-means [7], the fuzzy c-means algorithm (FCM) [8, 9] and the Gaussian mixture model (GMM) solved by Expectation-maximization (EM) algorithm [10]. The *K*-means algorithm clusters data by iteratively computing a mean intensity for each class and segmenting the image by classifying each pixel in the class with the closest mean. This algorithm belongs to the hard segmentation and cannot meet our requirement.

The fuzzy c-means algorithm generalizes the K-means algorithm [11], allowing for soft segmentations based on fuzzy set theory. For a joint-volume with N voxels, the voxel intensity vectors are denoted by $x_i (i = 1,2,\cdots,N)$. If K is the number of tissues (or materials), the fuzzy c-means is an iterative optimization that minimizes the cost function defined as follows:

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ik}^{m} \left\| x_i - \mu_k \right\|^2$$

where u_{ik} represents the membership of voxel x_i in the kth cluster, μ_k is the kth cluster center.

The Gaussian mixture model assumes that each voxel is composed by K component densities mixed together with K mixing coefficients. Each component density follows a Gaussian distribution. Based on statistical theory, the parameters are estimated by maximum likelihood (ML) and EM algorithm is used as an optimization method.

Both the fuzzy c-means algorithm and the Gaussian mixture model based method algorithm belong to the soft segmentation method. The fuzzy c-means estimates the parameters which minimize the distance from each voxel to the class centers. It uses only the distance objective function without any other information about the intensity distributions. In contrast, the method based on Gaussian mixture model uses the statistical theory to model each voxel's intensity, which is more reasonable to the real situation. In this chapter, we choose the Gaussian mixture model and estimate the ML parameters by EM algorithm.

All the clustering algorithms mentioned above do not directly incorporate spatial modeling. So when dealing with the data containing partial volume effect, these methods can lead to misclassification at the object boundaries because they rely only on the intensity distributions during the classification process. To understand misclassification, let us consider a situation where a dataset has three tissues A, B and C, with scalar values f(a), f(b) and f(c), respectively, such that f(a)< f(b)< f(c). Let us

assume that the tissues A and C touch each other, chances are very high that the boundary between A and C is classified to B. In addition, the lack of spatial information during classification will lead to noise sensitivity in inhomogeneous regions. Lakare [12] treated the sample locations in the partial volume area specially to solve this misclassification problem. When detecting a partial volume boundary, the author chose the closer material value (between the two component materials of the boundary) instead of the sampled value in order to avoid the misclassification. This method takes the partial volume effect into classification process, but the classification result is definite decisions at the partial volume boundary instead of the material probabilities and it only focused on the PVE problem without considering the noise sensitivity problem.

G. Kindlmann noted for intensity-only classification problems in his tutorial [1] that "histograms/scatter-plots entirely loose spatial information" and he asked if there would be "any way to keep some of it?". As described by Roettger et al. [4], spatial information is important, because a feature by definition is a spatially connected region in the volume domain with a unique position and certain statistical properties. They indicated that only using the statistical information of the scatter-plot will effectively ignore the most important part of a features definition.

Many researchers have realized the importance of spatial information for image segmentation (classification). Zhang et al. [13] proposed a novel hidden Markov random field (HMRF) model to integrate spatial information to Gaussian model based segmentation methods. Instead of using Markov random filed (MRF) as a general prior in Gaussian model based approach as other researchers did [14], the author proposed a Gaussian hidden Markov random field model and used an MRF-MAP approach to estimate class labels, while MAP was used to estimate the bias field which is only exist in MR images. The bias field doesn't exist in CT images and in addition this model estimates a definite class label for each pixel without the consideration of PVE, so that it doesn't meet the requirement of our situation. Tang et al. [15] proposed to use a multi-resolution Gaussian mixture model method for image segmentation in order to solve the noise sensitivity problem of Gaussian mixture model based method. The spatial information is implicitly contained in the higher level image. Chuang et al. [8] integrated the spatial information to fuzzy c-means algorithm by incorporating it into the membership function.

We also agree that the spatial information is very important for classification. In order to integrate spatial information to the Gaussian mixture model based vectorial image segmentation method, we proposed to involve a neighborhood weight to the classification process. To reach this goal, we need a neighborhood information descriptor. Lunstrom et al [16] proposed the Partial Range Histogram (PRH) concept, which is a way to describe the amount of a tissue within a local region. This gives us the hint to use this concept as a neighborhood descriptor. Inspired by this neighborhood description form, we propose a neighborhood weighted Gaussian mixture classification method with the purpose of getting a more accurate classification result.

3.2. Gaussian mixture model

For a joint-volume with *N* voxels, each voxel is a *n*-dimensional vector. The voxel intensity vectors are denoted by $x_i (i = 1,2,\cdots,N)$. If *K* is the number of tissues (or materials), we assumed that each voxel intensity *f*(*x*) is composed by *K* component densities mixed together with *K* mixing coefficients:

$$f(x) = \sum_{k=1}^{K} \alpha_k f_k(x) \tag{3.1}$$

where $f_k(x)$ denotes the *k*th component density and α_k denotes the mixing coefficient of the *k*th material. If $f_k(x)$ follows Gaussian distribution, the model of Eq. (3.1) becomes a Gaussian mixture model (GMM). The Gaussian mixture model has been widely applied on MR image segmentation [17-19].

The Gaussian distribution of the *k*th tissue class is denoted by $p_k(x|\Theta_k)$, which is governed by a set of parameters Θ_k. Given the parameters of all the classes, the probability distribution of each voxel can be described as follows:

$$p(x_i|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(x_i|\Theta_k) \tag{3.2}$$

where the parameters are $\Theta = (\alpha_1,\cdots,\alpha_K,\Theta_1,\cdots,\Theta_K)$ with the constraint that $\sum_{k=1}^{K} \alpha_k = 1$.

Maximum likelihood (ML) estimation is a common used method to find the probability distribution parameters. The log-likelihood expression for this density from the data *X* is given by:

$$\log(L(\Theta|X)) = \log\prod_{i=1}^{N} p(x_i|\Theta) = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \alpha_k p_k(x_i|\Theta_k)\right) \tag{3.3}$$

Typically, p_k is modeled by a Gaussian distribution with mean μ_k and covariance matrix Σ_k. That is:

$$p_k(x_i|\Theta_k) = p_k(x_i|\mu_k,\Sigma_k) = \frac{1}{\sqrt{\det(2\pi\Sigma_k)}} e^{-(x_i-\mu_k)^T \Sigma_k^{-1}(x_i-\mu_k)/2} \tag{3.4}$$

Finding the ML solution directly from Eq. (3.3) is difficult because it contains the log of the sum. The EM algorithm [20] is a good way to solve this problem. In the next section, we describe how to maximize the likelihood by EM algorithm.

3.3. EM algorithm

3.3.1. EM algorithm principle [20]

The EM algorithm is a general method of finding the ML estimate of an underlying distribution parameters form a given dataset when the data is incomplete or has missing values. The EM algorithm is very useful in this situation: if we assume

the existence of the missing (or hidden) values, the likelihood function can be simplified.

This algorithm is based on the idea that the observed data X is incomplete. We assume that a complete dataset exists $Z=(X,Y)$ and Y is the missing data. So that a joint density function is:

$$f(Z;\Theta)=f(X,Y;\Theta)=f(Y\mid X;\Theta)\cdot f(X;\Theta) \tag{3.5}$$

So

$$\log f(X;\Theta)=\log f(X,Y;\Theta)-\log f(Y\mid X;\Theta) \tag{3.6}$$

The ML estimation can be equal to the maximization of the right side of Eq. (3.6). X is the observed data, so it can be considered as a constant. Take the conditional expectation of Eq. (3.6), we get:

$$E[\log f(X;\Theta)\mid X]=E[\log f(X,Y;\Theta)\mid X]-E[\log f(Y\mid X;\Theta)\mid X] \tag{3.7}$$

If X is given, $\log f(X\mid\Theta)$ is a certain function. So that $E[\log f(X;\Theta)\mid X]\equiv\log f(X;\Theta)$. The right side of Eq. (3.7) is equal to the original function.

We assume that there is another parameter Θ' that isn't equal to Θ. According to Eq. (3.7), we can get this equation:

$$\begin{aligned}L(\Theta')-L(\Theta)=&E[\log f(X,Y;\Theta')\mid X]-E[\log f(X,Y;\Theta)\mid X]\\&-(E[\log f(Y\mid X;\Theta')\mid X]-E[\log f(Y\mid X;\Theta)\mid X])\end{aligned} \tag{3.8}$$

As Y is a random variable, we can define:

$$Q(\Theta',\Theta)\triangleq E[\log f(X,Y;\Theta')\mid X]=\int f(Y;\Theta\mid X)\log f(X,Y;\Theta')dY \tag{3.9}$$

$$K(\Theta',\Theta)\triangleq E[\log f(Y\mid X;\Theta')\mid X]=\int f(Y\mid X;\Theta)\log f(Y\mid X;\Theta')dY \tag{3.10}$$

Using Eqs. (3.9) and (3.10), Eq. (3.8) becomes:

$$L(\Theta')-L(\Theta)=Q(\Theta',\Theta)-Q(\Theta,\Theta)-(K(\Theta',\Theta)-K(\Theta,\Theta)) \tag{3.11}$$

According to the non-negative property of relative entropy, we can get this decision:

$$K(\Theta',\Theta)-K(\Theta,\Theta)=\int f(Y\mid X;\Theta)\log\frac{f(Y\mid X;\Theta')}{f(Y\mid X;\Theta)}dY\le 0 \tag{3.12}$$

Considering Eqs. (3.11) and (3.12), we can find that if $Q(\Theta',\Theta)-Q(\Theta,\Theta)\ge 0$, we can get $L(\Theta')-L(\Theta)\ge 0$. That is to say, if we want to find the ML estimation $\hat{\Theta}$, we can construct a series $\{\hat{\Theta}^t\}$ to satisfy that:

$$L(\hat{\Theta}^{t+1})\ge L(\hat{\Theta}^t),\forall t=0,1,2,\cdots \tag{3.13}$$

According to Eqs. (3.11) and (3.12), this series only need to satisfy the condition:

$$Q(\hat{\Theta}^{t+1},\hat{\Theta}^t)\ge Q(\hat{\Theta}^t,\hat{\Theta}^t),\forall t=0,1,2,\cdots \tag{3.14}$$

The EM algorithm process can be summarized as follows:

1) Initialize the parameters, get $\hat{\Theta}^0$.

2) E-step: Under the condition that the observed data *X* and the current parameters $\hat{\Theta}^t$ are known, compute the conditional expectation:

$$Q(\Theta, \hat{\Theta}^t) = \int f(Y; \hat{\Theta}^t \mid X) \log f(Y, X; \Theta) dY \tag{3.15}$$

3) M-step: Compute the optimization solution of $Q(\Theta, \hat{\Theta}^t)$ to update the parameters:

$$\hat{\Theta}^{t+1} = \arg\max_{\Theta} Q(\Theta, \hat{\Theta}^t) \tag{3.16}$$

4) Let $t \triangleq t+1$ and repeat 2) and 3) until the end of the iteration.

3.3.2. Finding ML estimation via EM algorithm [10]

The main part of EM algorithm is the two steps: E-step and M-step. The first step (E-step) is to find the appropriate function *Q*. This function is a conditional expectation to the unobserved random vector *Y*.

$$Q(\Theta, \Theta^t) = E_Y[\log P(X, Y \mid \Theta) \mid X, \Theta^t] \tag{3.17}$$

We assume that *Y* is the probability that the *i*th voxel is generated by the *k*th class. That is to say, when the *i*th voxel is generated by the *k*th class, $y_i = k$. In this case, Eq. (3.17) takes the form:

$$\begin{aligned}
&E_Y[\log P(X, Y \mid \Theta) \mid X, \Theta^t] \\
&= \sum_{y \in \Upsilon} \log[\prod_{i=1}^{N} \alpha_{y_i} p_{y_i}(x_i \mid \Theta_{y_i})] \cdot \prod_{j=1}^{N} p(y_j \mid x_j, \Theta^t) \\
&= \sum_{y \in \Upsilon} \sum_{i=1}^{N} \log(\alpha_{y_i} p_{y_i}(x_i \mid \Theta_{y_i})) \cdot \prod_{j=1}^{N} p(y_j \mid x_j, \Theta^t) \\
&= \sum_{y_1=1}^{K} \sum_{y_2=1}^{K} \cdots \sum_{y_N=1}^{K} \sum_{i=1}^{N} \log(\alpha_{y_i} p_{y_i}(x_i \mid \Theta_{y_i})) \cdot \prod_{j=1}^{N} p(y_j \mid x_j, \Theta^t) \\
&= \sum_{y_1=1}^{K} \cdots \sum_{y_N=1}^{K} \sum_{i=1}^{N} \sum_{k=1}^{K} \delta_{y_i,k} \log(\alpha_k p_k(x_i \mid \Theta_k)) \cdot \prod_{j=1}^{N} p(y_j \mid x_j, \Theta^t) \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} \log(\alpha_k p_k(x_i \mid \Theta_k)) \sum_{y_1=1}^{K} \cdots \sum_{y_N=1}^{K} \delta_{y_i,k} \prod_{j=1}^{N} p(y_j \mid x_j, \Theta^t)
\end{aligned} \tag{3.18}$$

This form looks very daunting, but it can be simplified:

$$\begin{aligned}
&\sum_{y_1=1}^{K} \cdots \sum_{y_N=1}^{K} \delta_{y_i,k} \prod_{j=1}^{N} p(y_j \mid x_j, \Theta^t) \\
&= \left(\sum_{y_1=1}^{K} \cdots \sum_{y_{i-1}=1}^{K} \sum_{y_{i+1}=1}^{K} \cdots \sum_{y_N=1}^{K} \prod_{j=1, j \neq i}^{N} p(y_j \mid x_j, \Theta^t) \right) \cdot p(k \mid x_i, \Theta^t) \\
&= \prod_{j=1, j \neq i}^{N} \left(\sum_{y_j=1}^{K} p(y_j \mid x_j, \Theta^t) \right) \cdot p(k \mid x_i, \Theta^t)
\end{aligned} \tag{3.19}$$

Since $\sum_{y_j=1}^{K} p(y_j \mid x_j, \Theta^t) = 1$, using Eqs. (3.18) and (3.19), Eq. (3.17) can be written as:

$$\begin{aligned} & Q(\Theta, \Theta^t) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} \log(\alpha_k p_k(x_i \mid \Theta_k)) p(k \mid x_i, \Theta^t) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} \log(\alpha_k) p(k \mid x_i, \Theta^t) + \sum_{i=1}^{N}\sum_{k=1}^{K} \log(p_k(x_i \mid \Theta_k)) p(k \mid x_i, \Theta^t) \end{aligned} \tag{3.20}$$

The next step is to maximize the function *Q*, which is called M-step. From Eq. (3.20) we can see that the term containing α_k and the term containing Θ_k can be maximized independently because they are not related.

As $\sum_k \alpha_k = 1$, we introduce the Lagrange multiplier λ and then solve the following equation:

$$\begin{aligned} & \frac{\partial}{\partial \alpha_k}\left[\sum_{i=1}^{N}\sum_{k=1}^{K} \log(\alpha_k) p(k \mid x_i, \Theta^t) + \lambda\left(\sum_k \alpha_k - 1\right)\right] = 0 \\ & \Rightarrow \sum_{i=1}^{N} \frac{1}{\alpha_k} p(k \mid x_i, \Theta^t) + \lambda = 0 \end{aligned} \tag{3.21}$$

Summing both sizes over *k*, we get that $\lambda = -N$ resulting in:

$$\alpha_k^{new} = \frac{1}{N}\sum_{i=1}^{N} p(k \mid x_i, \Theta^t) \tag{3.22}$$

Using Eq. (3.4), the Θ_k part of Eq. (3.20) becomes:

$$\begin{aligned} & \sum_{i=1}^{N}\sum_{k=1}^{K} \log(p_k(x_i \mid \Theta_k)) p(k \mid x_i, \Theta^t) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K}\left(-\frac{1}{2}\log(2\pi)^d - \frac{1}{2}\log(\det(\Sigma_k)) - \frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k)\right) p(k \mid x_i, \Theta^t) \end{aligned} \tag{3.23}$$

If *A* is a symmetric matrix, *X*, *a*, *b* are vectors, the following results from matrix algebra is useful to compute the derivatives of Eq. (3.23).

$$\frac{\partial}{\partial X}\left(X^T A X\right) = 2AX \tag{3.24}$$

$$\frac{\partial \det(A)}{\partial A} = \det(A)(A^{-T}) \tag{3.25}$$

$$\frac{\partial}{\partial A}\left(a^T A^{-1} b\right) = -A^{-T} a b^T A^{-T} \tag{3.26}$$

Using Eq. (3.24), take the derivative of Eq. (3.23) with respect to μ_k and set it equal to zero, we get:

$$\sum_{i=1}^{N} \Sigma_k^{-1}(x_i - \mu_k) p(k \mid x_i, \Theta^t) = 0 \tag{3.27}$$

Solve this equation, we obtain the estimate of μ_k:

$$\mu_k^{new} = \frac{\sum_{i=1}^{N} x_i \cdot p(k \mid x_i, \Theta^t)}{\sum_{i=1}^{N} p(k \mid x_i, \Theta^t)} \tag{3.28}$$

Similarly, using Eqs. (3.25) and (3.26) we can get the estimate of Σ_k:

$$\Sigma_k^{new} = \frac{\sum_{i=1}^{N} p(k \mid x_i, \Theta^t) \cdot (x_i - \mu_k^{new})(x_i - \mu_k^{new})^T}{\sum_{i=1}^{N} p(k \mid x_i, \Theta^t)} \tag{3.29}$$

Using Bayes' rule, we can compute:

$$p(k \mid x_i, \Theta^t) = \frac{\alpha_k^t p_k(x_i \mid \Theta_k^t)}{p(x_i \mid \Theta^t)} = \frac{\alpha_k^t p_k(x_i \mid \Theta_k^t)}{\sum_{j=1}^{K} \alpha_j^t p_j(x_i \mid \Theta_j^t)} \tag{3.30}$$

Eqs. (3.22), (3.28) and (3.29) are the update formula during iteration. Based on these equations, the estimation process can be summarized as in Fig. 3.1.

(Input: The vectorial volume $x_i (i = 1, 2, \cdots, N)$, the number of classes *K*.)

Step 1: Initialization of Θ^0 and α^0. Any classification method could be used, in our case we choose K-means[2].

Step 2: Calculate the prior probability by Eq. (3.30).

Step 3: Compute the new parameter data according to Eqs. (3.22), (3.28) and (3.29).

Step 4: Let $t \triangleq t + 1$ and repeat steps 2-3 until reaching the end condition.

Figure 3.1: Vectorial volume classification algorithm by Gaussian mixture model

[2] K-means decomposes the volume into *K* clusters with n_k voxels in each cluster. μ_k^0 is initialized by the center of gravity of cluster *k*. Σ_k^0 is initialized by the variance of cluster *k*. α_k^0 is initialized by the ratio of n_k to the total voxel number of the volume.

3.4. Proposed neighborhood weighted classification

3.4.1. Modified model with neighborhood information

The iteration formula described in section 3.3 didn't involve any spatial information about current voxel. As discussed in section 3.1, neighborhood information is one of the most important spatial information. If the iteration procedure takes the neighborhood effect into account, the classification result can be more reasonable.

The original model calculates the class probabilities according to Bayes' rule, which is described by Eq. (3.30). This calculation is based on intensity distributions without any neighborhood information. Usually the material is continuous, so that it is natural to have the idea that for each voxel, the probability of the *k*th class should be affected by the neighbors' *k*th class probabilities. According to this belief, we can integrate the neighborhood effect on the class distributions of the current voxel by modifying Eq. (3.30).

Due to the deducing process of EM algorithm and the natural continuous properties of the classes (materials), the class probability should obey the two rules:

1) $\sum_{k=1}^{K} p(k \mid x_i, \Theta^t) = 1,$

2) Current voxel's *k*th class probability magnifies if the neighbors' *k*th class probabilities tend to 1; current voxel's *k*th class probability decreases if the neighbors' *k*th class probabilities tend to 0.

According to the second rule, the neighborhood class distribution can be designed as a weight on the current class distribution, so that we designed the neighborhood weighted probability for the current voxel:

$$p(k \mid x_i, \Theta^t) = \frac{\alpha_k^t W_{ik}^t p_k(x_i \mid \Theta_k^t)}{\sum_{j=1}^{K} \alpha_j^t W_{ij}^t p_j(x_i \mid \Theta_j^t)} \tag{3.31}$$

Where

$$W_{ik} = \frac{\sum_{n=1}^{|N_i|} p\left(k \mid x_{ni}, \Theta^t\right)}{|N_i|} \tag{3.32}$$

N_i is a set of neighborhood of the *i*th voxel. $|N_i|$ denotes the number of voxels in a set N_i. x_{ni} denotes the *n*th neighbor's intensity of the *i*th voxel.

Now we briefly prove that the designed formula Eq. (3.31) can satisfy the two rules. When taking Eq. (3.31) into the left side of the first rule, it equals to 1 so that Eq. (3.31) can rule 1) naturally.

In order to prove rule 2), we assume that for the *k*th class of the *i*th voxel, there exist two neighborhood weights W_1 and W_2 with: $W_1 > W_2 > 0$, the corresponding probabilities calculated by Eq. (3.31) are denoted by p_1 and p_2 respectively. If we can prove that $p_1 > p_2$, the second rule can be satisfied. We denote $M \triangleq \alpha_k^t p_k(x_i \mid \Theta_k^t)$, $N \triangleq \sum_{j=1\cdots K, j\neq k} \alpha_j^t W_{ij}^t p_j(x_i \mid \Theta_j^t)$. When the variable is the neighborhood weight, *M* and *N*

are constant and $M \geq 0$, $N \geq 0$. We can rewrite Eq. (3.31) for W_1 and W_2 respectively:

$$p_1 = \frac{MW_1}{MW_1 + N}, \; p_2 = \frac{MW_2}{MW_2 + N}$$

so that

$$p_1 - p_2 = \frac{MN(W_1 - W_2)}{(MW_1 + N)(MW_2 + N)}$$

When *M* and *N* are constant and positive, we can deduce that if $W_1 > W_2 > 0$, then $p_1 > p_2$. When the condition is $W_{ik} = 0$, the probability $p(k \mid x_i, \Theta^t) = 0$. That is to say, the weighted probability according to Eq. (3.31) is a monotonously increasing function to the neighborhood weight W_{ik}. With this property, the rule 2) can also be satisfied.

The new class distribution formula is conformable to the two rules and integrates the neighborhood information to the current voxel's class distribution during iteration. For each iteration step, the class distribution will be amended by the neighbors' class distribution information. So that through this weighted formula, the neighborhood information is taken into account to the classification process.

3.4.2. Description of the algorithm

The EM solution formula for the proposed neighborhood weighted Gaussian mixture model is summarized as follows:

E-step:

$$p(k \mid x_i, \Theta^t) = \frac{\alpha_k^t W_{ik}^t p_k(x_i \mid \Theta_k^t)}{\sum_{j=1}^{K} \alpha_j^t W_{ij}^t p_j(x_i \mid \Theta_j^t)} \tag{3.33}$$

M-step:

$$\alpha_k^{t+1} = \frac{1}{N} \sum_{i=1}^{N} p(k \mid x_i, \Theta^t) \tag{3.34}$$

$$\mu_k^{t+1} = \frac{\sum_{i=1}^{N} x_i \cdot p(k \mid x_i, \Theta^t)}{\sum_{i=1}^{N} p(k \mid x_i, \Theta^t)} \tag{3.35}$$

$$\Sigma_k^{t+1} = \frac{\sum_{i=1}^{N} p(k \mid x_i, \Theta^t) \cdot (x_i - \mu_k^{t+1})(x_i - \mu_k^{t+1})^T}{\sum_{i=1}^{N} p(k \mid x_i, \Theta^t)} \tag{3.36}$$

$$W_{ik}^{t+1} = \frac{\sum_{n=1}^{|N_i|} p\left(k \mid x_{ni}, \Theta^t\right)}{|N_i|} \tag{3.37}$$

Based on these equations, the estimation process we implemented is described in Fig. 3.2.

For each element vector of the input series, the aim is to find its class distributions. From the iteration process, we can see that this algorithm is not limited in applying on vectorial volume. According to the spatial dimension (denoted by *D*) of the input series $x_i(i=1,2,\cdots,N)$ with *N* elements, the shape of the vectorial image to be classified can be a line (*D*=1), an image (*D*=2) or a volume (*D*=3). The difference is that the shape of N_i in Eq. (3.37) should match the dimension of the input series. Here we only take the nearest neighbors into account with: *D*=1, $|N_i|$=2; *D*=2, $|N_i|$=8; *D*=3, $|N_i|$=26. The neighborhood shape and topology could be modified according to the real practical situation.

(Input: The vectorial volume $x_i(i=1,2,\cdots,N)$, the number of classes *K*.)

Step 1: Initialization of Θ^0, α^0 and W^0. Any classification method could be used, in our case we choose K-means. Using Eq. (3.32) to initialize the neighborhood weight for each voxel.

Step 2: Calculate the prior probability by Eq. (3.33).

Step 3: Compute the new parameter data according to Eqs. (3.34), (3.35), (3.36) and (3.37).

Step 4: Let $t \triangleq t+1$ and repeat steps 2-3 until reaching the end condition.

Figure 3.2: Vectorial volume classification algorithm by proposed neighborhood weighted Gaussian mixture model

3.5. Experiments and discussions

In order to illustrate the classification results, we use vectorial images to test our algorithm instead of volumes. Comparing to images, the only difference for the vectorial volumes is that the shape of neighborhood should change accordingly, as mentioned in section 3.4.2. In order to avoid the local maxima, the algorithm is initialized as follows: for the synthetic data, we initialize the center points with a random data near the global maximum; for the practical situation, this initialization can be implemented by picking a point in each material region manually.

We evaluate our algorithm on both synthetic and real data. The effect of neighborhood range choice is also discussed.

3.5.1. Evaluation on synthetic data

3.5.1.1. Evaluation on two-dimensional vectorial image

We start the experiments with a simplest case that each pixel in the vectorial image contains only two elements. That is to say, each pixel is a vector with two elements. The test vectorial image (Fig. 3.3) is constructed as following:

The first channel image (Fig. 3.3 left) is composed by two homogeneous regions in which we add Gaussian noise: ($\mu=5,\sigma=1.2$) and ($\mu=10,\sigma=1$) respectively. The second channel image (Fig. 3.3 right) follows the same scheme with: ($\mu=15,\sigma=2$) and ($\mu=5,\sigma=0.9$) respectively.

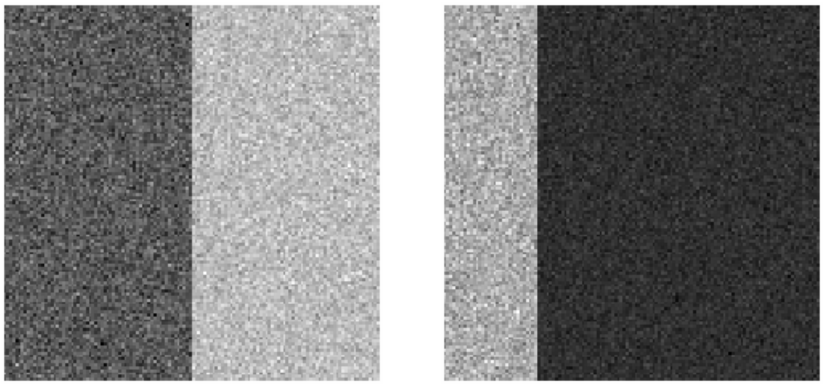


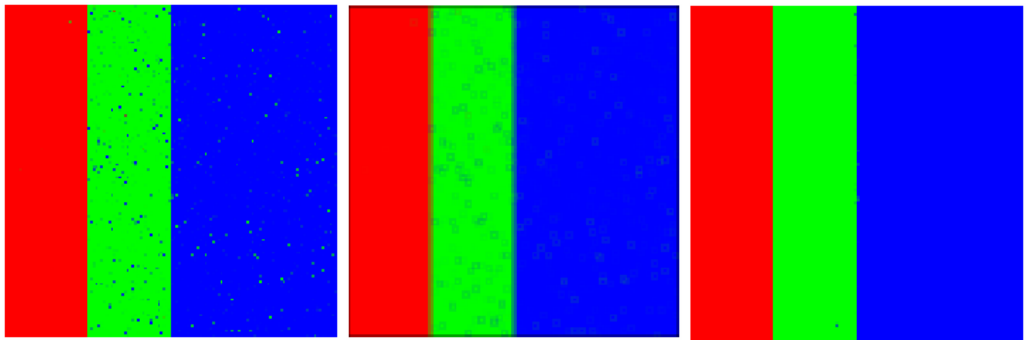
Figure 3.3: Synthetic data. Each image is one channel of the vectorial image.

The combination of these two channels leads to a vectorial image with three classes. According to the proposed algorithm described in Section 3.4, the input number of classes is K=3.

The classification on synthetic data is performed and the result is shown in Fig. 3.4. Each pixel of the result image is formed by this formula:

$$C(x_i) = \sum_{k=1}^{K} C_k \, p(k \mid x_i, \Theta) \tag{3.38}$$

Where $C(x_i)$ is the color assigned to the ith pixel and C_k is the color we assigned to the kth class.



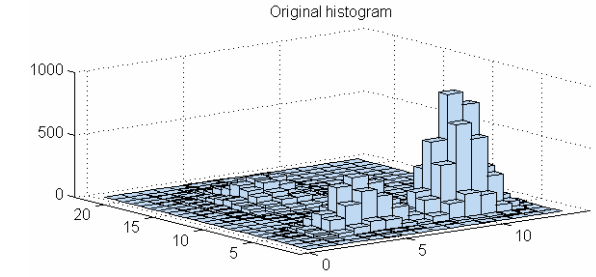
(a) (b) (c)

Figure 3.4: Classification result of the synthetic data. (a): the Gaussian mixture method; (b): directly smooth the class decisions with neighborhood after classification with the Gaussian mixture method; (c): our method.

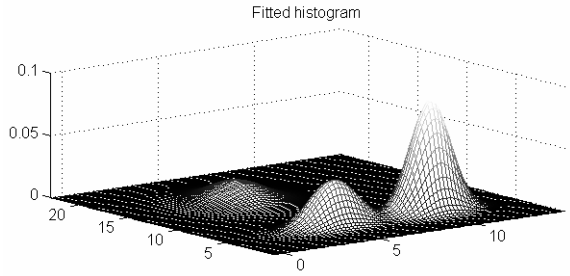
Fig. 3.4(a) is the classification result with the original Gaussian mixture model. We can notice that the final regions are not homogeneous as expected because of the noise. The reason is that the method relies only on the intensity distribution (histogram). The classification progress is a direct mapping from intensity to classes so that the noise areas are assigned an incorrect class distribution. In order to demonstrate that our method is not a simple class decisions smoothness. We use the neighborhood described in section 3.4.2 to smooth the class decisions after classification with the Gaussian mixture model and the result is illustrated in Fig.

3.4(b). We can see that although the noises are faded, the edges are blurred at the same time, which is not corresponding to the real situation. When we integrate the neighborhood information into the iteration procedure, we get the result in Fig. 3.4(c). It is obvious that the regions are more homogeneous and the classification process is less affected by the noise.

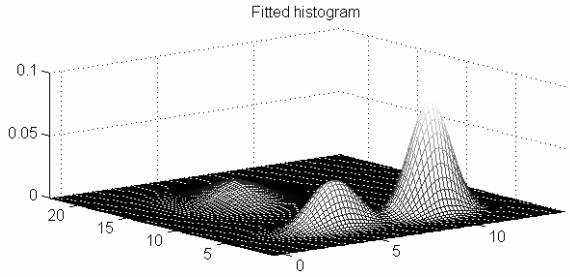
In the practical situation, histograms are considered as an approximation of the class distribution. Fig. 3.5 illustrates the histogram of the synthetic vectorial image and the corresponding classified class distributions. The original image's histogram is shown in Fig. 3.5(a) and the Gaussian mixture classification result is illustrated in Fig. 3.5(b). Our neighborhood weighted classification result is shown in Fig. 3.5(c). From these figures, we can see that the classification fits the original histogram very well. We cannot see and even measure significant differences between Fig. 3.5 (b) and (c). That is to say, the neighborhood weighted method doesn't change the intensity distributions. The effect of our proposed method is to amend the class decision by its neighborhood on each pixel during the iteration procedure, while keeping the global intensity distribution nature.



(a) Original histogram



(b) Classification without neighborhood



(c) Classification with neighborhood

Figure 3.5: Histogram of the vectorial image. (a): Original histogram; (b): The result distribution after classification by the Gaussian mixture model; (c): The result distribution after classification with our method.

From the analysis before, we can see that taking the neighborhood into account doesn't change the class distribution parameters. When we go back to Fig. 3.4, we can see that the difference between these results comes from the calculation of $p(k \mid x_i, \Theta)$. Fig. 3.4(a) is calculated according to Eq. (3.30), that is, without any neighborhood weight. Fig. 3.4(b) and (c) are both calculated according to Eq. (3.33) with the neighborhood weight W_{ik}, but the apparent results are quite different. The weight W_{ik} in Fig. 3.4(b) is calculated according to the final result of the class parameters estimation process, in contrast, W_{ik} in Fig. 3.4(c) is iteratively estimated during the EM algorithm, as presented in section 3.4.2. Comparing Fig. 3.4(b) and (c), we can easily reach the conclusion that when taking the estimation of neighborhood weight into the EM algorithm, the classification result is more conformable to the reality.

3.5.1.2. Experiments on three-dimensional vectorial image

In the real situation, there are usually more than two elements contained in each pixel of the vectorial image. We also did some experiments on a three-elements vectorial image. The test vectorial image (Fig. 3.6) is constructed as following:
The first channel image (Fig. 3.6 left) is composed by two homogeneous regions in which we add Gaussian noise: ($\mu = 3, \sigma = 1.5$) and ($\mu = 10, \sigma = 1.5$) respectively. The second channel image (Fig. 3.6 middle) follows the same scheme with: ($\mu = 15, \sigma = 1.5$) and ($\mu = 5, \sigma = 1.5$) respectively. The third channel image (Fig. 3.6 right) follows also the same scheme with: ($\mu = 2, \sigma = 1.5$) and ($\mu = 8, \sigma = 1.5$) respectively.

The three images (Fig. 3.6) form a three-elements vectorial image, in which each channel of the vector forms an independent image. The combination of these three channels leads to a vectorial image with six classes. According to the proposed algorithm described in Section 3.4, the input number of classes is K=6.

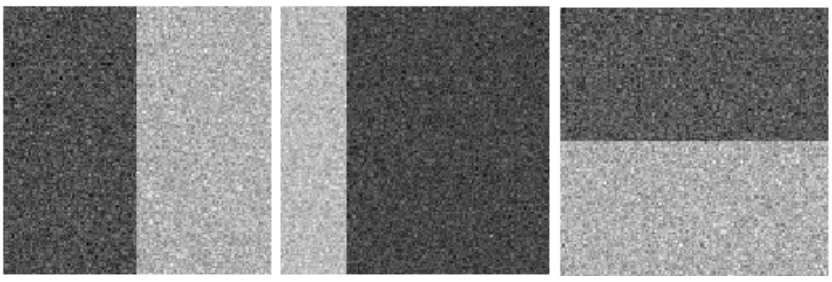
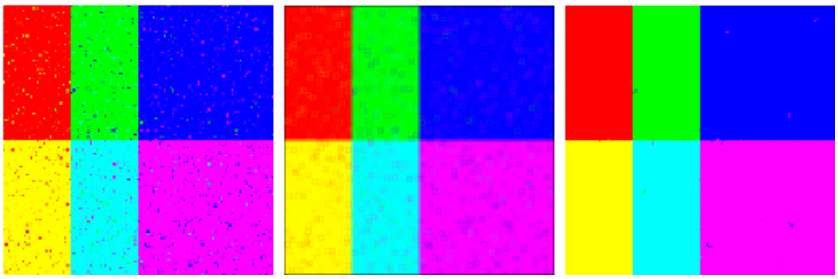


Figure 3.6: Synthetic data. Each image is one channel of the vectorial image.

Similar to the classification of two-elements vectorial image discussed in section 3.5.1.1, the corresponding results formed by Eq. (3.38) are illustrated in Fig. 3.7. The results demonstrate that our proposed algorithm also works well for three-elements vectorial image.

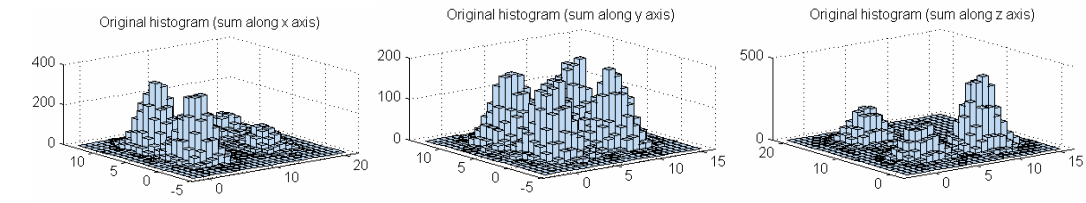
It is difficult to illustrate the histogram and the result intensity distribution directly for three-elements vectorial image because the histogram has four axes. We solve this problem by calculating the intensity distribution summation along one axis to reduce the total number of axis. The results are shown in Fig. 3.8. Fig. 3.8(a) illustrates the intensity distribution summation along three axes of the original image.



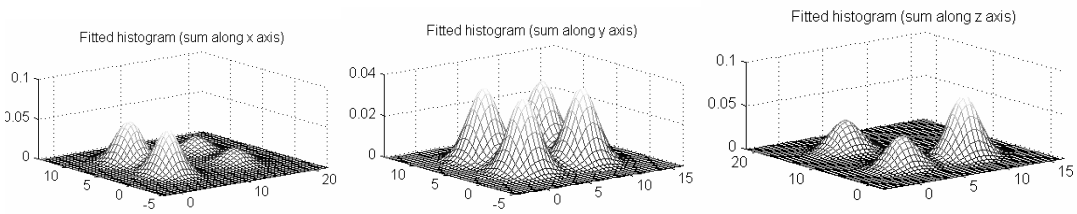
(a) (b) (c)

Figure 3.7: Classification result of the synthetic data. (a): the Gaussian mixture method; (b): directly smooth the class decisions with neighborhood after classification with the Gaussian mixture method; (c): our method.

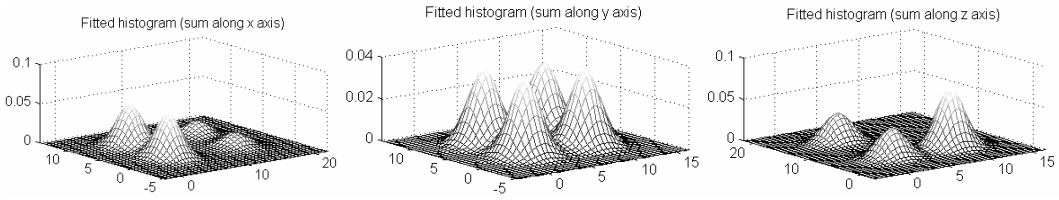
Fig. 3.8(b) is the classification result of Gaussian mixture classification and Fig. 3.8(c) illustrates the neighborhood weighted classification results. These histograms also demonstrate that the neighborhood weighted method doesn't change the intensity distribution of the final result.



(a) Original histogram



(b) Classification without neighborhood

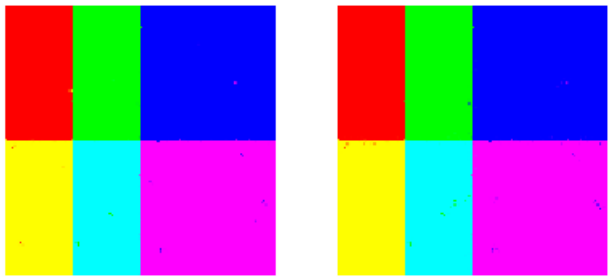


(c) Classification with neighborhood

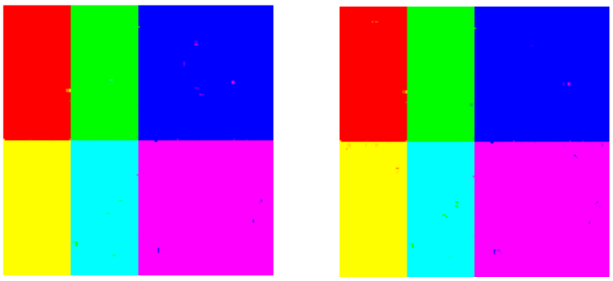
Figure 3.8: Intensity distribution along one axis. (a): Original image; (b): The results after classification by the Gaussian mixture model; (c): The results after classification with our method.

We also did some experiments while changing the size of the neighborhood in order to see its effect on the classification results. The results are illustrated in Fig. 3.9. We cannot see significant differences between these results so that the shape of the neighborhood plays a tiny part in the classification procedure for this synthetic data.

In our case, we choose the standard 8-connectivity neighborhood. For the other spatial dimensions (line, volume, etc.), the neighborhood should change accordingly as described in section 3.4.2.



(a) 8-connectivity (b) 24-connectivity



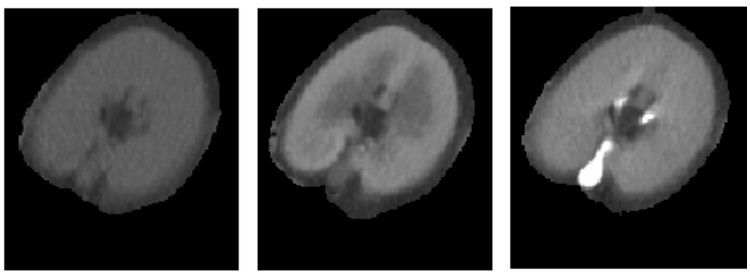
0	0	1	0	0
0	1	2	1	0
1	2	0	2	1
0	1	2	1	0
0	0	1	0	0

(c) 4-connectivity (d) special shape

Figure 3.9: Classification results with different neighborhood shapes.

3.5.2. Application on real data

After the evaluation on synthetic data, we performed the methods on the real data obtained after the registration of three CT acquisitions. Fig. 3.10 shows one slice of the registered vectorial volume, which is composed by three channels: (a), acquisition before contrast medium injection; (b), immediately after injection; (c), ten minutes after injection.

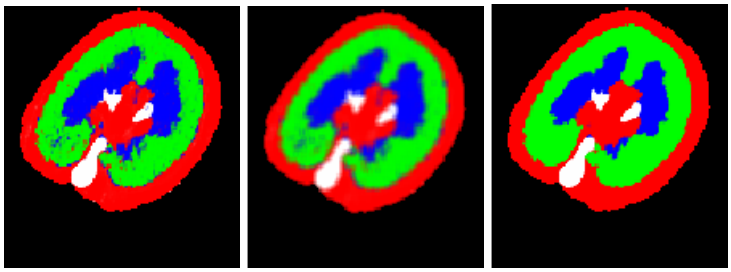


(a) (b) (c)

Figure 3.10: One slice of the kidney volume after registration.

We expect to classify our volume into four classes: fat, renal cortex, renal medulla and collecting system. With K=4, the classification result formed by Eq. (3.38) is shown in Fig. 3.11. It effectively demonstrates our conjecture. While taking the neighborhood information into classification procedure (Fig. 3.11(c)), the anatomical structures are better delineated into homogeneous regions: fat (red), renal cortex

(green), renal medulla (blue) and collecting system (white). The simple class decisions smoothness (Fig. 3.11(b)) blurs the image and cannot get an accurate classification result especially on the tissue border.



(a) (b) (c)

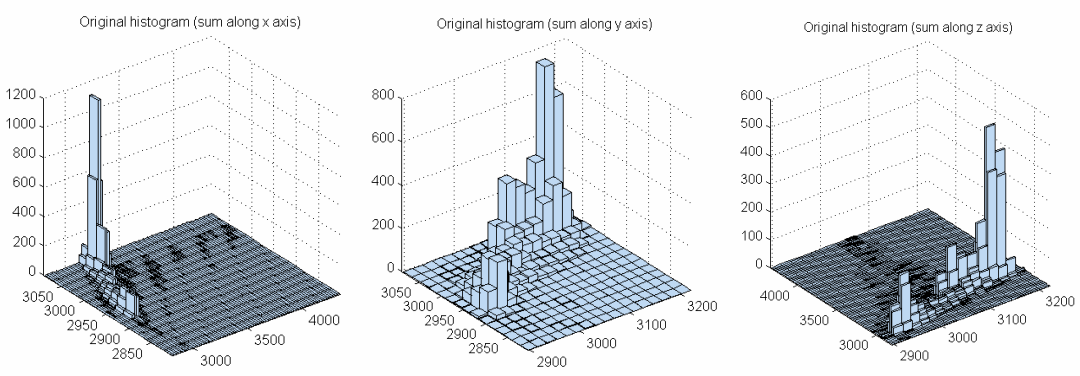
Figure 3.11: Classification result of the real. (a): the Gaussian mixture method; (b): directly smooth the class decisions with neighborhood after classification with the Gaussian mixture method; (c): our method.

Similar to the experiments on synthetic data, we also illustrate the histograms by integrating the intensity distributions along one axis (Fig. 3.12). We can see that the fitted histogram agree with the original histogram very well. The difference between the classified intensity distribution with and without neighborhood is tiny. The results confirm that the proposed neighborhood weighted classification method gives the pixels' class probabilities concerning the neighborhood information while maintaining the global intensity distribution.

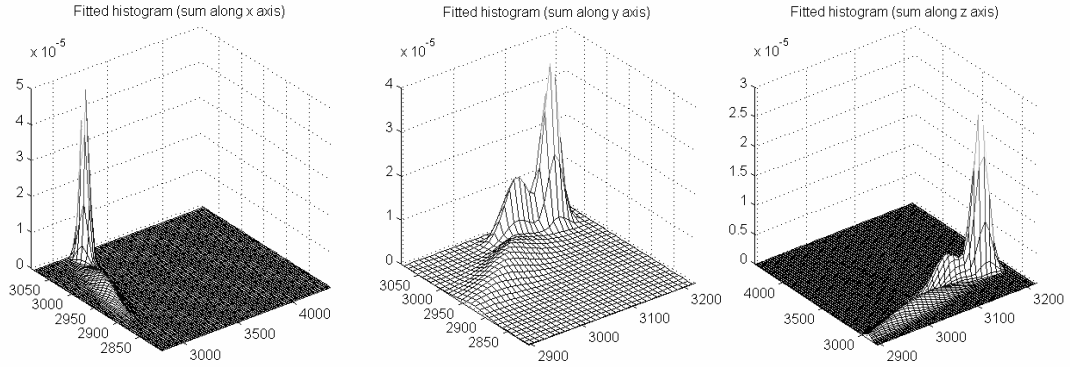
3.5.3. Discussions

From the above results, we can reach the conclusion that the Gaussian mixture model based method has the ability to classify vectorial image with the aim of illustrating the anatomical structures. Because of the inhomogeneity of the acquisitions and the partial volume effects, the result of the intensity-only method has some misclassification area, especially the renal cortex and the renal medulla because of their close intensity range, which is shown in Fig. 3.11(a). In order to illustrate this phenomenon more clearly, the corresponding first order derivate of the result probabilities along one cut line (represented in white) is shown in Fig. 3.13. Because of the white background color, the collecting system (represented in white originally) is represented in black line in the probabilities derivate figure (Fig. 3.13), the other materials are represented according to the colors originally assigned to them.

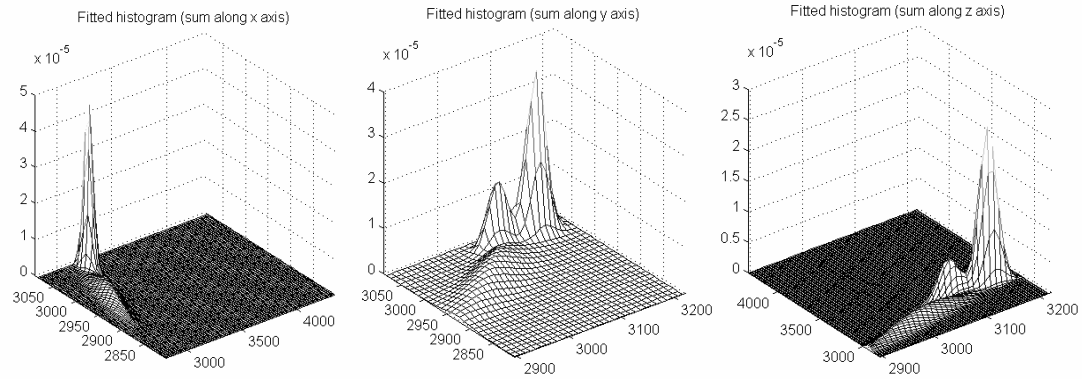
In fig. 3.13(a), according to the probabilities first order derivate, we can clearly see that the regions are not separated because there are some inhomogeneous regions, e.g. in the renal medulla (between index [30, 40] on the line) or partial volume effect on tiny object, e.g. the collecting system (between index [70, 80] on the line). While taking the neighborhood information into the iteration process, the results are improved significantly, as shown in Fig. 3.13(b). We can see that inside one material region, the proposed method gives a more homogeneous decision. The proposed method considers the intensity and the position of one pixel simultaneously so that it can give a more reasonable classification result. While comparing Fig. 3.13 (a) and (b), we can see that the proposed method has the effect of less sensitive to inhomogeneous region, while giving a better class distribution.



(a) Original histogram

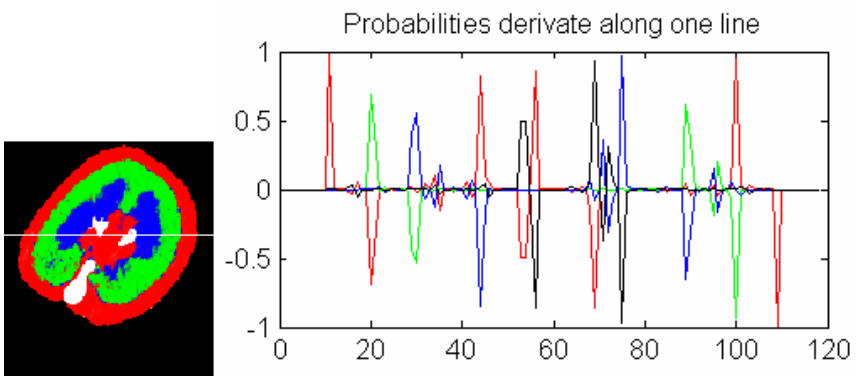


(b) Classification without neighborhood

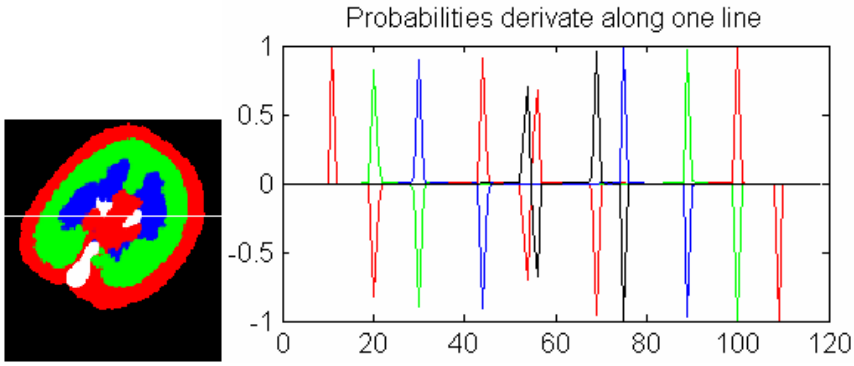


(c) Classification with neighborhood

Figure 3.12: Intensity distribution along one axis. (a): Original image; (b): The results after classification by the Gaussian mixture model; (c): The results after classification with our method.



(a)



(b)

Figure 3.13: Probability first order derivate along one cut line. (a): the Gaussian mixture method; (b): our method

3.6. Conclusions

In this chapter, Gaussian kernel functions are used in a statistical classifier to have the ability of classifying vectorial images or volumes. A neighborhood weighted method is proposed. The model is that the voxels' intensity vectors follow the Gaussian mixture distribution and the classes distributions on each voxel are affected by its neighbors' class probability distributions so that a neighborhood weight is used to describe this property. The neighborhood information is integrated to the classification process by amending the voxel's class distributions at each iteration step.

Experiments on both synthetic and real data are performed. The results show that this improvement on Gaussian mixture model is less affected by noise and gives better classification results. Usually, a nearest neighborhood is enough in practical use. The experiments also demonstrate that the expansion of the neighborhood range makes tiny effect in the final classification results. The experimental results demonstrate that the proposed method gives more reasonable class distributions for each pixel while keeping the global intensity distribution.

References

[1] G. Kindlmann, "IEEE visualization 2001 tutorial 5, presentation 3: transfer functions for direct volume rendering," presented at Visualization'01, San Diego, CA, 2001.

[2] W. Cai and G. Sakas, "Data intermixing and multi-volume rendering," *Computer Graphics Forum*, vol. 18, pp. 359-368, 1999.

[3] E. A. Firle and M. Keil, "Multi-volume visualization using spatialized transfer functions. Gradient- vs. multi-intensity-based approach," presented at CARS 2007, Berlin, Germany, 2007, pp. S121-S123.

[4] S. Roettger, M. Bauer, and M. Stamminger, "Spatialized transfer functions," presented at EUROGRAPHICS-IEEE VGTC Symposium on Visualization, Leeds, UK, 2005, pp. 271-278.

[5] B. Wilson, E. B. Lum, and K.-L. Ma, "Interactive multi-volume visualization," presented at the International Conference on Computational Science (ICCS), Amsterdam, Holand, 2002, pp. 102-110.

[6] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, vol. 2, pp. 315-337, 2000.

[7] G. B. Coleman and H. C. Andrews, "Image segmentation by clustering," *Proceedings of the IEEE*, vol. 67, pp. 773-785, 1979.

[8] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation," *Computerized Medical Imaging and Graphics*, vol. 30, pp. 9-15, 2006.

[9] M. S. Yang, Y. J. Hu, K. C. Lin, and C. C. Lin, "Segmentation techniques for tissue differentiation in MRI of ophthalmology using fuzzy clustering algorithms," *Magn Reson Imaging*, vol. 20, pp. 173-9, 2002.

[10] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," International computer science institute, Berkeley, California ICSI-TR-97-021, 1998.

[11] T. N. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans on signal processing*, vol. 40, pp. 901-914, 1992.

[12] S. A. lakare, "Ray based exploration of volumetric data," in *Computer Science*, vol. PhD: Stony Brook University, 2004.

[13] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm," *IEEE Trans Med Imaging*, vol. 20, pp. 45-57, 2001.

[14] T. Kapur, E. L. Grimson, R. Kikinis, and W. M. Wells, "Enhanced spatial priors for segmentation of magnetic resonance imagery," presented at MICCAI'98, Cambridge, MA, 1998, pp. 148-157.

[15] Y. G. Tang, D. Liu, and X. P. Guan, "Multi-resolution image segmentation based on Gaussian mixture model," *Journal of Systems Engineering and Electronics*, vol. 17, pp. 870-874, 2006.

[16] C. Lundstrom, P. Ljung, and A. Ynnerman, "Extending and simplifying transfer function design in medical volume rendering using local histograms," presented at EUROGRPHICS, Leeds, UK, 2005, pp. 263-270.

[17] W. M. Wells, W. L. Grimson, R. Kikinis, and F. A. Jolesz, "Statistical intensity correction and segmentation of MRI data," presented at 3rd Conf. Visualization Biomed. Computing (VBC'94), Rochester, MN, 1994, pp. 13-24.

[18] W. M. Wells, W. L. Grimson, R. Kikinis, and F. A. Jolesz, "Adaptive segmentation of MRI data," *IEEE Trans Med Imaging*, vol. 15, pp. 429-42, 1996.

[19] R. Guillemaud and M. Brady, "Estimating the bias field of MR images," *IEEE Trans Med Imaging*, vol. 16, pp. 238-51, 1997.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1-38, 1977.

Chapter 4: Vectorial volume visualization techniques

For scalar volume rendering, there exist many volume visualization methods [1]. There are relatively less researches focused on the rendering of vectorial volume composed by multi-volumes, that is, volumes acquired by scanning an object with multiple modalities or the same modality at several period of time. But because of the development of scan techniques and registration methods [2], this vectorial volume data becomes arisen in medical visualization.

Most of the existing methods [3-5] for this vectorial volume visualization are based on the intermixing of the component volumes at one certain step in the rendering pipeline, these spatial-aligned volumes are still considered individually. But all the volumes are from the same patient, they should be combined together as a vectorial volume. We have presented a neighborhood weighted Gaussian mixture classification method to get the class distributions of the vectorial volume. With the help of that vectorial volume statistical classification method, we can consider this vectorial volume as an integrative volume and render it with several rendering techniques.

The rendering techniques are divided into two categories: surface rendering and volume rendering. Surface rendering techniques are relatively easier to implement in our situation. But its disadvantage is that a geometrical presentation should be obtained first and the final image relies mostly on the generation quality of the surface models. Direct volume rendering methods can get the final image directly from the volumes, but in our situation, the difficulties appear in gradient calculation, transfer function design etc. We focus on the solvent of these difficulties and propose several rendering techniques for classified vectorial volumes in this chapter.

4.1. Introduction

In this section, we first outline our research context of this chapter. The representation of the results from the previous work gives us a classified volume. The specificity of this volume is that each voxel is a vector and the tissues memberships are assigned to each voxel by a classification method. To display this volume, we need a suitable vectorial volume visualization algorithm. Vectorial volume visualization techniques are expanded from scalar volume visualization techniques. For the investigation of vectorial volume visualization algorithms, we first briefly review the scalar volume visualization algorithms then the existing vectorial visualization algorithms are analyzed in comparison to our proposed method. Finally, the outline of our work in this chapter is given.

In chapter 1 we have shown that the general visualization framework can be adapted to our situation (Fig. 4.1).

The next step is now the graphical encoding of the general model. In order to distinguish all the tissues in the final image, we need to assign a color and a transparency value for each material. For the *k*th material, the user will assign a color C_k and an opacity value α_k (denotes the transparency of the material). The opacity has such properties: $\alpha_k = 1$ implies that the *k*th material is completely opaque, and $\alpha_k = 0$ implies that it is completely transparent.

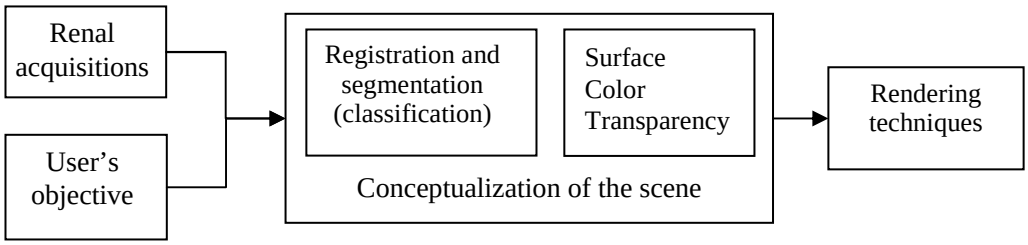


Figure 4.1: General framework of a visualization tool for the kidney surgery preoperative planning

As referred in Fig. 4.1, these steps conclude the process of the conceptualization of the scene and now the next step is to find appropriate rendering techniques for the final visualization.

Scalar volume visualization

In this chapter, we focus on the rendering techniques for vectorial volumes. But because the vectorial volume visualization techniques are expanded from scalar volume visualization techniques we first introduce the scalar volume visualization.

For the visualization of scalar volumes, most of the algorithms can be classified into two categories: surface rendering algorithms and direct volume rendering algorithms [6]. Surface rendering algorithms first extract the surface representations from the volume data and then graphic techniques are used to render the extracted geometric primitives. Volume rendering algorithms directly get the final image from the volume data without going though an intermediate surface extraction step.

For surface rendering algorithms, a surface shell is extracted from the data. The surface is typically approximated as triangular mesh which can be passed as geometry to a rendering process. The classical approach to surface extraction is the *Marching Cubes* algorithm, proposed by Lorensen and Cline [7]. This algorithm assumes that the data is on a structured grid and then extracts the iso-value surface within a unit cube and processes the cubes independently one after the other.

Unlike the surface approximation method, direct volume rendering convey an entire 3D dataset in a 2D image directly. There are many research concerned about volume rendering [1, 8, 9]. The process of constructing an image from a volumetric dataset using direct volume rendering can be summarized by the following steps [10]:

1) **Data traversal:** The positions where samples will be taken from the volume are determined.
2) **Sampling:** The dataset is sampled at the chosen positions. The sampling points typically do not coincide with the grid points, and so interpolation is needed to reconstruct the sample value.
3) **Gradient computation:** The gradient of the data is often needed, in particular as input to the shading component. Gradient computation requires additional sampling.
4) **Transfer function:** The sampled values are mapped to optical properties, typically color and opacity value. The transfer function is used to visually distinguish materials in the volume.

5) **Shading and illumination:** Shading and illumination effects can be used to modulate the appearance of the samples. The three-dimensional impression is often enhanced by gradient-based shading.

6) **Compositing:** The pixels of the rendered image are computed by compositing the optical properties α_s and the colors C_s of the sample points according to the volume rendering integral.

Vectorial volume visualization

For the vectorial volume visualization methods, we shall first distinguish between vectors of physical quantities (such as flow and strain) and vectors that store a list of voxel attributes. There is a large body of literature to visualize the former, but it is out of our discussion range. We focus on the latter, that is, volumes which are composed by voxels of attribute vectors. Here, in particularly, are the volumes acquired by scanning an object at different periods, which is similar to multi-modal volumes. Each voxel of the vectorial volume contains a vector formed by the patient's acquisitions from different scan times.

Based on the fundamental algorithms used for scalar volume visualization, a few literatures did some research about the vectorial visualization techniques [3-5]. These authors consider the vectorial volume as a collection of separate volumes and they mix the vector's components at one certain step of the scalar volume rendering pipeline.

Cai and Sakas [5] classified the methods according to the levels where the data intermixing occurs. Three levels were defined: *image level intermixing*, *composition level intermixing* and *illumination model level intermixing* (Fig. 4.2).

- The simplest mixing technique is image level intermixing. It consists to render each volume separately as a scalar dataset and then to blend the result images according to some weighting function that possibly includes the z-buffer of opacity channel. This method doesn't require a modification of the volume renderer but it loses the depth ordering information.
- The composition level intermixing method solved this problem. For each voxel of each volume, the opacity and color are estimated according to the voxel value and the illumination model. These opacities and colors are then intermixed at each compositing step, thus preserving the depth information.
- A third method is illumination model level intermixing. The volume samples are combined before colors and opacities are computed.

Although these intermixing methods can render the vectorial volume, they all consider the space aligned volumes individually. In addition, these methods mix the data at different steps of the volume rendering pipeline so that the surface rendering techniques cannot be applied at all.

In our case, the volumes are taken from the same patient at different contrast periods so that they can be considered together. The intermixing level can then occur before the rendering pipeline. We will call this level *acquisition level intermixing* (Fig. 4.2). The acquisition intermixing step is realized by the neighborhood weighted mixture Gaussian classification method we proposed in Chapter 3. After this classification, we get the material probability distributions on each grid of the volume. As introduced later, both surface rendering and volume rendering techniques for

scalar volume visualization can be adapted to the vectorial volume visualization with this acquisition level intermixing method.

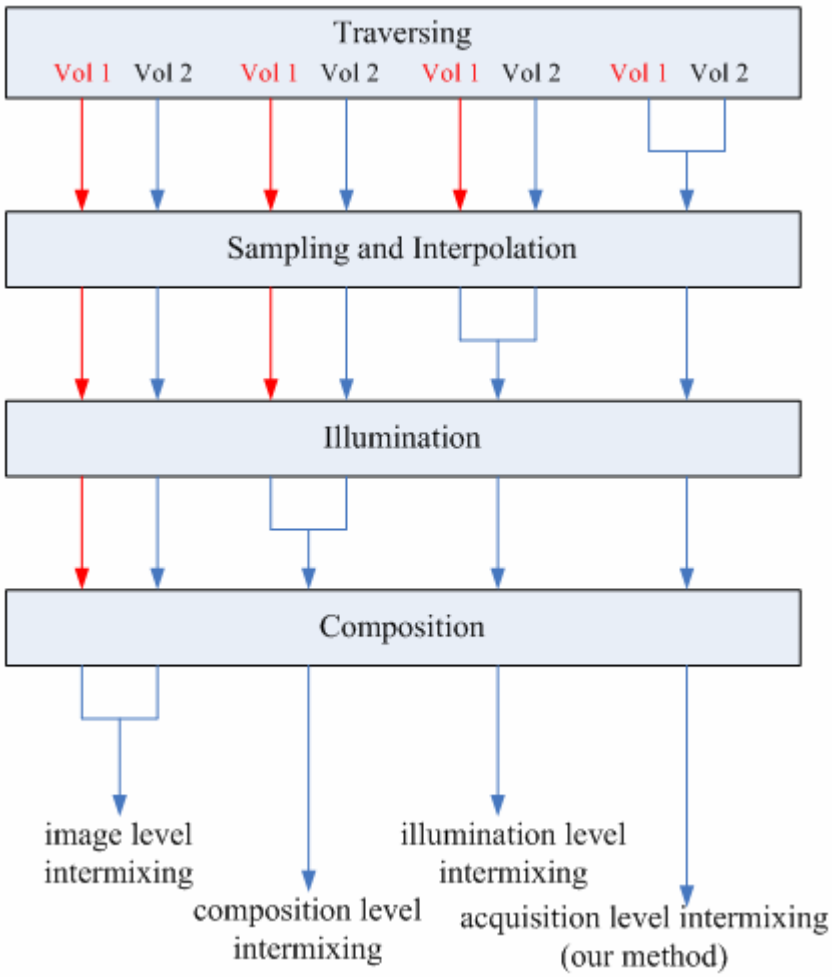


Figure 4.2: Rendering pipeline for different intermixing levels

Outline of our work

Both surface rendering and volume rendering techniques can be expanded to the vectorial volume visualization in the acquisition level intermixing method. Surface rendering method applied in our situation is a simple multi-object expansion, which will be introduced in detail in section 4.2.

Volume rendering is relatively complex because some of its steps cannot be simply expanded for the intermixed vectorial volume. According to the volume rendering procedure, the difficulty of the expansion from scalar volume to vectorial exists in the gradient computation and the transfer function design because the sample values are intensity vectors and material probabilities instead of scalar values. We focus on solving these problems in the direct volume rendering section of this chapter (section 4.3). Finally, our discussions and conclusions are given.

4.2. Surface rendering method

For scalar volume, surface rendering techniques approximate a surface by some geometrical primitives, most commonly triangles, which can then be rendered using conventional graphics accelerator hardware. A surface can be defined by applying a binary decision function $B(v)$ to the volumetric data, where $B(v)$ evaluates to 1 if the value v is considered part of the object, and evaluates to 0 if the value v is part of the background. The surface is then contained in the region where $B(v)$ changes from 0 to 1. When $B(v)$ is a step function: $B(v) = 1, \forall v \geq v_{iso}$, where v_{iso} is called the *iso-value*, the resulting surface is called the *iso-surface* [1]. The *Marching Cubes* algorithm [7] was developed to approximate an iso-valued surface with a triangle mesh.

This surface extraction cannot be applied on the vectorial volume directly because the vectorial *iso-value* is difficult to define. But the class distributions we get after applying the vectorial classification method are relatively separated, as shown in Fig. 4.3. Each material distribution can be treated as an independent volume. Each material distribution volume is closed to a binary volume with the value range [0, 1] (especially at the border) instead of always 1 in a binary volume. We can get the surfaces of each class and then render them in the same scene as multiple objects so that the materials can be merged in the final image.

Figure 4.3: One slice of the material distributions, from left to right: fat, renal cortex, renal medulla and collecting system.

The value range of the material distribution is [0, 1]. From Fig. 4.3 we can see that this data range only happens at the border and inside the object the material probabilities tend to 1, so that we choose 0.5 as the *iso-value* to extract the surfaces of the objects. The algorithm is summarized as follows:

1) Set the *iso-value* to 0.5 and extract surfaces from each material distribution volume.
2) Assign material color and transparency to the corresponding surface.
3) Render the surfaces in the same scene by the graphical rendering techniques.

The results get from step 1) and 2) are illustrated in Fig. 4.4. In this figure, we do not consider fat because it's not useful for illustrate the anatomical structure of the kidney. We can see that the three materials are relatively independent from each other. At the border of two materials, the algorithm detects the surface for each of them respectively. That is to say, the algorithm cannot separate the surface inside and outside of the object and all of the borders are detected as surfaces. The three surfaces are rendered as three different objects with transparency properties and the final merged image is illustrated in Fig. 4.5. From the results, we can see that the multi-

object solution is practical for this situation. The advantage of this method is that the final image can be rendered very fast after the surface extraction. The disadvantage is that the surfaces should be extracted first and the volumes are reduced to the boundaries of materials and all the other information is lost.

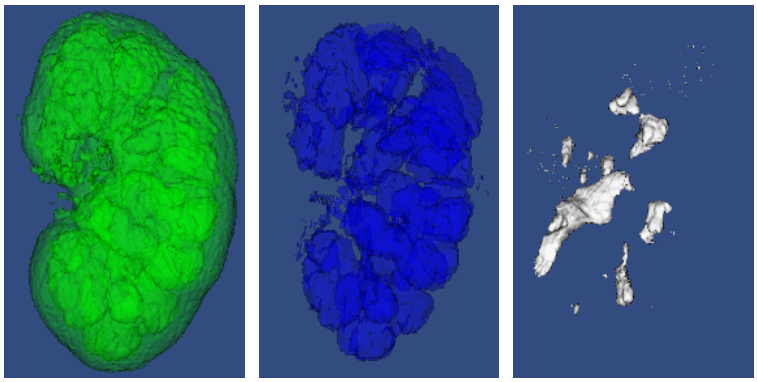


Figure 4.4: Surface extraction result of the material probability volumes, from left to right: renal cortex, renal medulla and collecting system.

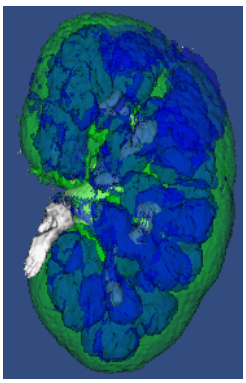


Figure 4.5: Merged scene rendered by semi-transparent surface rendering technique.

4.3. Direct volume rendering method

4.3.1. Introduction

Direct volume rendering is a visualization technique to convey an entire 3D dataset in a 2D image directly. The direct volume rendering process expressed by Hadwiger et al. [10] has been reviewed in section 4.1, in which transfer function is an important step to visually distinguish materials. Drebin et al. [11] proposed a direct volume rendering technique to visualize a scalar volume composed by several materials, which is similar to our situation, but the authors didn't pay much attention on the boundaries of materials. Kindlmann and Durkin [12] indicated the importance of transfer functions and analyzed the boundary properties. According to their analysis, the authors proposed to assign the opacity by a function of both data value and gradient magnitude. But the analysis has a significant assumption: the features of interest in the scalar volume are the boundary regions between areas of relatively homogeneous material. Considering these volume rendering methods for multi-materials, we proposed two volume rendering methods based on ray casting which enhance the boundaries by the design of transfer function.

4.3.2. Volume rendering framework

The framework of the ray casting rendering process is shown in Fig. 4.6. The sample positions depend on the direction of the casted rays. As introduced before, the input of this rendering pipeline is the material probabilities on each voxel and the material properties (color and opacity value) assigned by the user. Recall that the material probabilities on grids are gotten by the neighborhood weighted classification method presented in chapter 3.

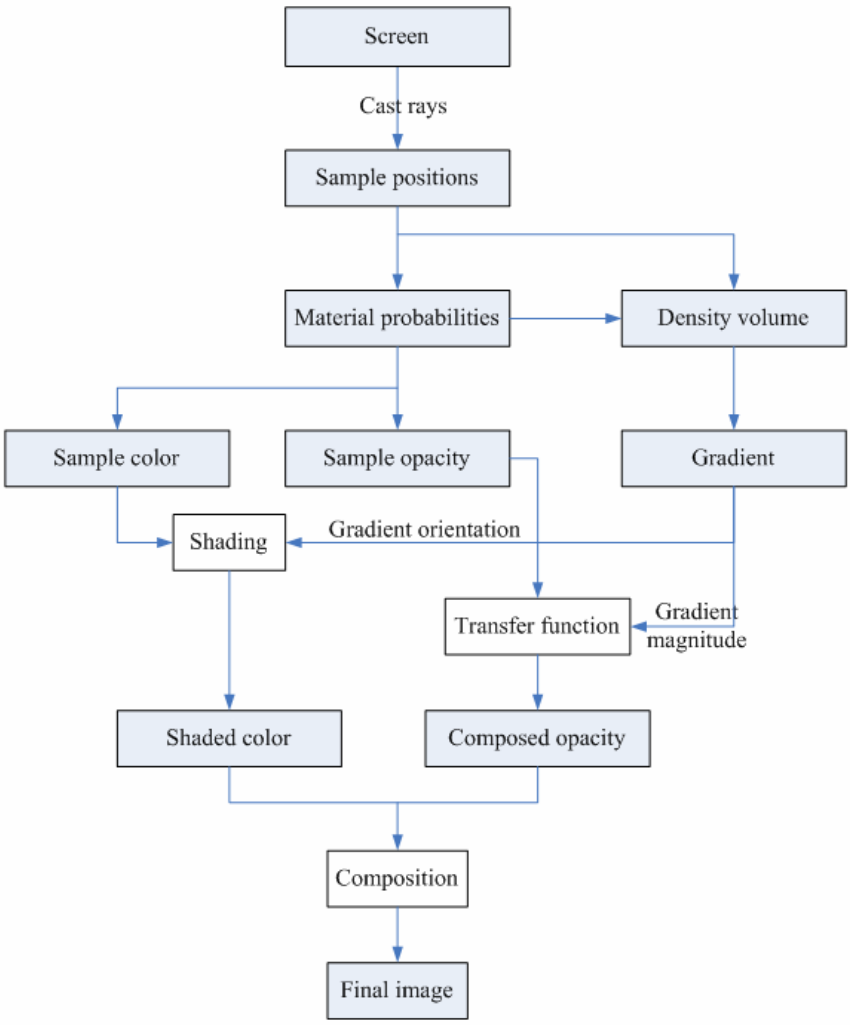


Figure 4.6: Volume rendering framework

The material probabilities on the sample points are achieved by interpolation method. For gradient calculation, Drebin et al. [11] proposed to form a density volume by assigning a density value to each material and then composing the densities weighted by the materials' probabilities. We apply a similar idea but use the opacity value instead of an extra density assignment, which will be expressed in detail in section 4.3.3.

From the sampled material probabilities and the assigned material properties, we get the sample color C_s and sample opacity α_s for the composition step of the rendering pipeline. The sample color and opacity are given by a transfer function. Concerning the opacity, the transfer function can have two roles [8]: (a) assign to a

specific voxel the tissue transparency and (b) enhance the surfaces by increasing the opacities in the boundary areas and decreasing them in homogeneous regions. Different transfer functions will lead us to different direct volume rendering methods, which will be described in detail in section 4.3.4. Kindlmann [13] summarized the transfer functions and indicated that gradient magnitude is a useful second dimension for transfer function design to enhance surfaces. We take this idea and adapt it to our practical situation.

Besides the transfer function, shading is another important issue in the rendering pipeline. Shading effects can be used to modulate the appearance of the samples. We apply the widely used Phong shading model [14] to calculate the shaded color.
The final step in the volume rendering process is the composition. It constitutes the optical foundation of the method. We will express it in detail in section 4.3.5. Finally, in section 4.3.6 we illustrate some experimental results and do some comparison.

4.3.3. Gradient calculation

Drebin et al. [11] proposed to form a density volume and calculate the gradient. The density volume is formed by the material properties. A density characteristic ρ_k is assigned for the *k*th material and then the density is formed by the following formula for each grid:

$$D = \sum_{k=1}^{K} p_k \rho_k \tag{4.1}$$

We can see that the greater ρ_k is, the more important the *k*th material is. If ρ_k equals to zero, the *k*th material will disappear in the final image.

As referred in section 4.1, the material opacity α has such properties: $\alpha = 1$ implies that the material is completely opaque, and $\alpha = 0$ implies that it is completely transparent. The material opacity has the similar effect as the density characteristic ρ so that we can use the opacity α to replace ρ to form the density volume. This replacement can reduce the input parameters of the rendering pipeline so that it can simplify the user input because the two properties: opacity α and the density ρ can be confused. The formula is as follows:

$$D = \sum_{k=1}^{K} p_k \alpha_k \tag{4.2}$$

-1	0	1
-2	0	2
-1	0	1

Figure 4.7: 2D Sobel operator

Our experimental dataset in this chapter is almost noise free because of the efficient classification method proposed in Chapter 3. We choose the Sobel operator for the gradient calculation because it provides the best edge quality for noise absence

edges comparing to other operators [15]. For illustration, Fig. 4.7 shows the mask of 2D Sobel operator in one direction. The 3D Sobel operator is a special expansion of the 2D case. Denote the gradient vector by $\mathbf{G}$, and $\mathbf{G}=(G_x,G_y,G_z)$. G_x, G_y and G_z are the directional gradient in *x*, *y* and *z* axis direction respectively. The corresponding formula to calculate the gradient is as follows:

$$\begin{aligned} G_x = &(2\cdot(D_{x+1,y,z}-D_{x-1,y,z})+(D_{x+1,y+1,z}-D_{x-1,y+1,z}) \\ &+(D_{x+1,y-1,z}-D_{x-1,y-1,z})+(D_{x+1,y,z+1}-D_{x-1,y,z+1})+(D_{x+1,y,z-1}-D_{x-1,y,z-1}))/12 \end{aligned} \tag{4.3}$$

$$\begin{aligned} G_y = &(2\cdot(D_{x,y+1,z}-D_{x,y-1,z})+(D_{x+1,y+1,z}-D_{x+1,y-1,z}) \\ &+(D_{x-1,y+1,z}-D_{x-1,y-1,z})+(D_{x,y+1,z+1}-D_{x,y-1,z+1})+(D_{x,y+1,z-1}-D_{x,y-1,z-1}))/12 \end{aligned} \tag{4.4}$$

$$\begin{aligned} G_z = &(2\cdot(D_{x,y,z+1}-D_{x,y,z-1})+(D_{x+1,y,z+1}-D_{x+1,y,z-1}) \\ &+(D_{x-1,y,z+1}-D_{x-1,y,z-1})+(D_{x,y+1,z+1}-D_{x,y+1,z-1})+(D_{x,y-1,z+1}-D_{x,y-1,z-1}))/12 \end{aligned} \tag{4.5}$$

The normal direction is the gradient vector normalized by its magnitude:

$$\mathbf{N}=(G_x/|\mathbf{G}|,G_y/|\mathbf{G}|,G_z/|\mathbf{G}|) \tag{4.6}$$

where $|\mathbf{G}|$ denotes the magnitude of $\mathbf{G}$, with $|\mathbf{G}|=\sqrt{G_x^2+G_y^2+G_z^2}$.

4.3.4. Transfer function

Transfer function is applied to assign one color and one opacity value to each sample point comprising the volume dataset in order to visually discern the several materials in the final image. For our situation, we introduce two methods to assign the colors and opacities to the sample points.

4.3.4.1. Class decision method

The input is the classified probabilities, it is natural to consider making a material decision for each sample and then to assign the corresponding material properties to the sample point, which is called *class decision method*.

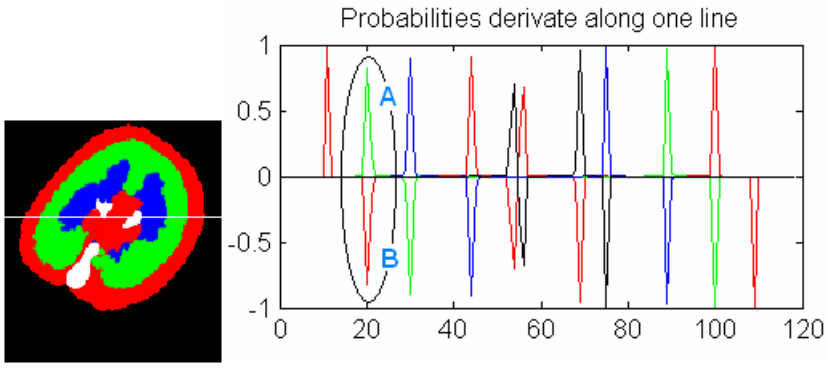


Figure 4.8: Probability first order derivate along one cut line. The circled region means the line goes from material *B* (red part) to material *A* (green part).

Recalling the first derivate analysis of the classified probabilities (Fig. 4.8), we can see the border of two materials very clearly. At the boundary region, the positive derivate indicates that the line is going inside the material and the negative derivate

indicates going out of the material. The first derivate tends to zero when the line passes inside of the materials. When we cast a ray into the volume, we calculate the first derivate along the ray. With the analysis before, we can easily distinguish the inside of the one material and the borders of materials. We can also get the information that we are going from material *B* to material *A* if the first derivate of *A* is positive and the first derivate of *B* is negative, as illustrated in Fig. 4.8. Denote the directional first derivate of *k*th material as $f'(k)$, and the color and opacity of the *k*th material as C_k and α_k respectively, The sample color C_s and the sample opacity α_s are given by:

$$C_s = C_p, \quad \alpha_s = \alpha_p \cdot f'(p), \quad \text{where } p = \arg\max_k f'(k) \tag{4.7}$$

With this formula, the inside of the materials will be discarded because the first derivate tends to zero. From Fig. 4.8 we can see that the highest first derivate appears at the border of two materials which will give a high opacity according to Eq. 4.7. This formula defines a class decision for the boundary of two materials, so that it is called class decision method.

4.3.4.2. Composed color and opacity

Unlike the class decision method, the color and opacity for the sample position can be gotten from the material probabilities directly by multiplying the color and opacity assigned to that material by the probability of each material. The sample color C_s and sample opacity α_s are given by:

$$C_s = \sum_{k=1}^{K} p_k C_k, \quad \alpha_s = \sum_{k=1}^{K} p_k \alpha_k \tag{4.8}$$

where p_k denotes the probability of the *k*th material and the number of materials is *K*.

As mentioned before, during the rendering process, the regions of interest are boundaries between materials and the transfer function is an efficient tool to express the boundaries. Eq. (4.8) has given out the color and opacity of the sample position, but it doesn't have the ability to highlight any boundaries. The composed opacity has the same formula as the density volume (Eq. (4.2)). Fig. 4.9 illustrates the relationship between the sample location and the first derivative of the density volume. The first derivate is actually just the gradient magnitude [12] (the gradient computed on the densities *D*). We can see that the gradient magnitude magnifies at the material boundaries and equals to zero at the interior of one material, which is corresponding to the separated materials' first derivative (Fig. 4.8).

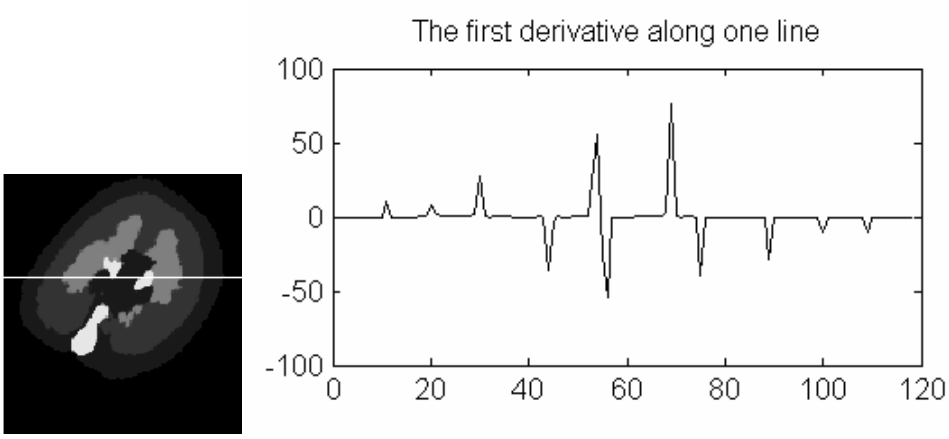


Figure 4.9: The first derivative of the density volume

Usually the gradient magnitude is very sensitive to the noise, but due to the efficiency of neighborhood weighted classification method, the result of the classified material is almost homogeneous. As the density volume is constructed according to the material opacities, the gradient magnitude agrees with the material opacity values. If two materials have similar opacities, the boundary between them will have small gradient magnitude; contrarily, if the two materials are quite different, the boundary between them will have a big gradient magnitude. The gradient magnitude can be considered as the "importance" of a boundary surface. If we use the gradient magnitude as an opacity mask, all the boundaries will appear in the final image according to their "importance".

The weighted opacity is described by the following formula:

$$\alpha_s = \left(\sum_{k=1}^{K} p_k \alpha_k \right) \cdot \left| \mathbf{N_s} \right|' \tag{4.9}$$

where $\left| \mathbf{N_s} \right|'$ is the normalized gradient magnitude of the sample position, it is given by:

$$\left| \mathbf{N_s} \right|' = \frac{\left| \mathbf{N_s} \right| - \left| \mathbf{N} \right|_{\min}}{\left| \mathbf{N} \right|_{\max} - \left| \mathbf{N} \right|_{\min}} \tag{4.10}$$

where $\left| \mathbf{N} \right|_{\max}$ and $\left| \mathbf{N} \right|_{\min}$ denote respectively the global maximum and minimum gradient magnitude of the whole volume.

4.3.5. Composition

The final step of volume rendering is composition, which has two kinds of direction: front-to-back and back-to-front. The advantage of the front-to-back composition order is that it can terminate the ray as soon as the accumulated ray opacity reaches a threshold close to full opacity. The reported benefit for this early ray termination is about 50% [9].

The fundamental process for creating a volume rendering image is based on simplified models of the real physical realism [16]. These optical models describe how a ray of light is affected when traveling through the volume. To compute an image, the effects of the optical properties must be integrated continuously along each

viewing ray. Among all the optical models, the *absorption plus emission model* is the most common one in direct volume rendering.

Let a and b denote the entry and exit points of the ray, I_0 be the light entering from the background, $T(u, v)$ be an aggregate of the transparency between u and v, $g(u)$ specifies the emission at a point u along the ray. The optical model can be described as follows:

$$I(b) = I_0 T(a,b) + \int_a^b g(u)T(u,b)du \tag{4.11}$$

The first term accounts for the absorption of light as the ray passes through the volume and the second term captures the emission from the inside of the volume, which is also affected by the volume.

This description assumes both the volume and the mapping to optical properties to be continuous. In practice, of course, the evaluation of the volume rendering integral is usually calculated numerically, together with several additional approximations. Max [16] also gave out the numerical calculation method after the description of the optical models. According to the integration sequence, both the back-to-front and front-to-back compositing algorithm can be deduced from the optical model (Eq. 4.11).

The ray is divided into n small segments. For the ith segment the emission contribution becomes a single color C_i. the transparency T_i is usually denoted by the opposite property *opacity*: $\alpha_i = 1 - T_i$. Assume that the enter color of the ith segment is C_{in} and the enter opacity is α_{in}. After the integration of the ith segment, the result color is C_{out} and the result opacity is α_{out}.

The back-to-front compositing formula is as follows:

$$C_{out} = C_{in}(1-\alpha_i) + C_i\alpha_i \quad i = 1, 2, \cdots, n \tag{4.12}$$

The front-to-back compositing formula is:

$$\left.\begin{aligned} C_{out}\alpha_{out} &= C_{in}\alpha_{in} + C_i\alpha_i(1-\alpha_{in}) \\ \alpha_{out} &= \alpha_{in} + \alpha_i(1-\alpha_{in}) \end{aligned}\right\} \quad i = n, n-1, \cdots, 1 \tag{4.13}$$

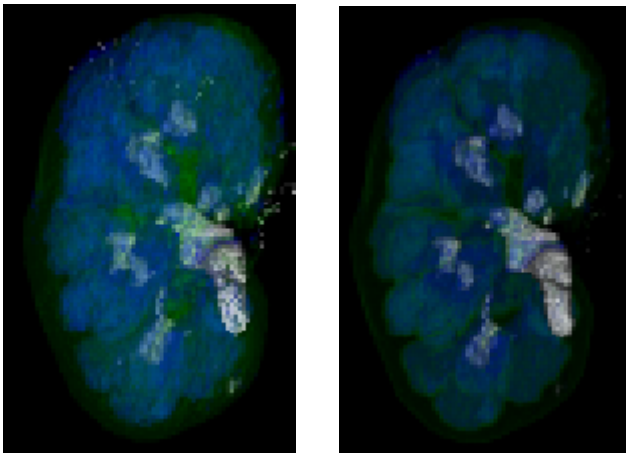
Comparing Eqs. (4.12) and (4.13), we can see that during the front-to-back compositing process, the cumulative opacity α magnified continuously. When α tends to 1, it means that this ray tends to be totally opaque and the following ray segments will have no more effect on the final pixel value, so that we can stop the integration process. Due to its ability of earlier termination, the front-to-back compositing algorithm gets a widely application.

4.3.6. Experimental results

According to the framework described in Fig. 4.6, we did some experiment on the classified volume. The input is the material probabilities on each grid and the materials' colors and opacities. The "fat" material is useless for the observer, so a totally transparency property (opacity equals to zero) is assigned to it.

Fig 4.10 shows the difference between the two transfer function design methods: Fig. 4.10(a) class decision method and Fig. 4.10(b) composed color and opacity. Comparing these two methods we can state the following remarks. On Fig. 4.10(a), we can see that the class decision method can better discriminate the different material;

there is less color confusion in the rendering result. This is the result that the boundaries between the materials lie on the position who gives the max first derivate of the material probabilities and that we use this information to make a material decision on each sample point and the corresponding material color and opacity is taken as the sampled color and opacity. But the decision making process is a 0-1 procedure. In contrast, on Fig. 4.10(b), the final result appears more continual because of the composed color and opacity but the boundaries are composed by two materials.



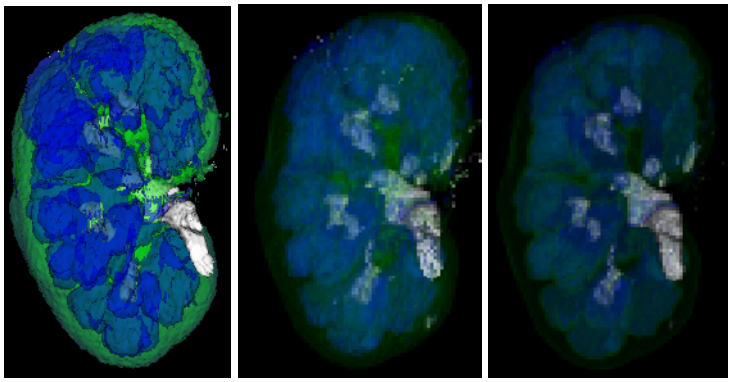
(a) (b)

Figure 4.10: Different transfer functions. (a): rendering result with class decision; (b): rendering result with composed color and opacity

4.4. Discussions and conclusions

Two categories of rendering techniques: surface rendering and direct volume rendering are introduced for the vectorial volume visualization in this chapter. The final experimental results are illustrated together in Fig. 4.11.

Surface rendering technique is relatively easier to apply in our situation, but the result (Fig. 4.11(a)) illustrates surfaces only and the geometry primitives should be extracted first. Direct volume rendering technique can get the final image directly from the classified material probabilities, but the rendering results rely much on the transfer function design. Different transfer functions will give different results. Two transfer functions are proposed: class decision method (Fig. 4.11(b)) and composed color and opacity method (Fig. 4.11(c)). According to the analysis of the first derivate, we propose to the gradient (first derivate) weighted opacity design method for both of the two transfer functions. The rendering result of the composed color and opacity is smoother than the class decision method. The experimental results demonstrate that both of the two direct volume rendering methods can highlight the boundaries with only a little user interaction (assignment of the material properties).



(a) (b) (c)

Figure 4.11: Comparison of different rendering techniques: (a), surface rendering; (b), volume rendering with class decision; (c), volume rendering with composed color and opacity.

In this chapter, several rendering techniques are used to visualize the classified result gotten by the classification method in chapter 3. Considering the classification and visualization process, we use the classification method to mix the component volumes of the vectorial volume. Then several rendering techniques are applied on the intermixed result. Comparing to other vectorial volume visualization methods, our method mixes the volumes at the acquisition intermixing level, which combines the acquisitions into a vector volume instead of several separated scalar volumes. The first step of this acquisition intermixing level visualization method has been discussed in chapter 3 and this chapter focuses on several following rendering techniques. Both of the surface rendering and direct volume rendering techniques are adapted to our situation. Two kinds of transfer design methods for direct volume rendering are implemented and compared. The comparison of these methods are given and discussed.

References

[1] A. Kaufman and K. Mueller, "Overview of volume rendering," in *The visualization Handbook*, C. R. Johnson, C. D. Hansen, and C. R. Johnson, Eds.: Academic Press, 2004.

[2] J. B. Maintz and M. A. Viergever, "A survey of medical image registration," *Med Image Anal*, vol. 2, pp. 1-36, 1998.

[3] E. A. Firle and M. Keil, "Multi-volume visualization using spatialized transfer functions. Gradient- vs. multi-intensity-based approach," presented at CARS 2007, Berlin, Germany, 2007, pp. S121-S123.

[4] B. Wilson, E. B. Lum, and K.-L. Ma, "Interactive multi-volume visualization," presented at the International Conference on Computational Science (ICCS), Amsterdam, Holand, 2002, pp. 102-110.

[5] W. Cai and G. Sakas, "Data intermixing and multi-volume rendering," *Computer Graphics Forum*, vol. 18, pp. 359-368, 1999.

[6] T. T. Elvins, "A survey of algorithms for volume visualization," presented at SIGGRAPH '92, Chicago, IL, 1992, pp. 194-201.

[7] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface reconstruction algorithm," *Computer Graphics*, vol. 21, pp. 163-169, 1987.

[8] M. Levoy, "Display of surfaces from volume data," *IEEE Comp Graph & Appl*, vol. 8, pp. 29-37, 1988.

[9] M. Levoy, "Efficient ray tracing of volume data," *ACM Trans. on Graphics*, vol. 9, pp. 245-261, 1990.

[10] M. Hadwiger, J. M. Kniss, C. Rezk-salama, D. Weiskopf, and K. Engel, *Real-time volume graphics*: A. K. Peters, Ltd., 2006.

[11] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," presented at SIGGRAPH'88, Atlanta, GA, 1988, pp. 65-74.

[12] G. Kindlmann and J. W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," presented at IEEE Symp on Volume visualization, Research Triangle Park, NC, USA, 1998, pp. 79-86.

[13] G. Kindlmann, "Transfer functions in direct volume rendering: design, interface, interaction," presented at SIGGRAPH'02 Course Notes, San Antonio, TX, 2002.

[14] B. Phong, "Illumination for computer generated pictures," *Communications of the ACM*, vol. 18, pp. 311-317, 1975.

[15] H. Trichili, M.-S. Bouhlel, N. Derbel, and L. Kamoun, "A survey and evaluation of edge detection operators application to medical images," presented at IEEE Conf. on Systems, Man and Cybernetics (SMC), 2002, pp. 4 pages.

[16] N. Max, "Optical models for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 99-108, 1995.

Chapter 5: Mesh simplification

For surface-based volume visualization methods, triangle meshes are often used to represent the object surfaces. But the total number of triangles used to represent object often largely exceeds the capacity of real-time rendering of graphics hardware. One natural way to solve it is to simplify the mesh models, replacing the original object with respectively fewer faces while trying to keep its main characteristics.

The simplification metric is a key issue of a simplification algorithm. In this chapter, two new simplification metrics based on surface moments and volume moments are proposed, which take the difference between the moments defined by the original mesh and those of the simplified mesh as the objective function. Edge collapse scheme is implemented as mesh simplification procedure. For a given maximum moment order and a required number of triangles, the optimal mesh with a minimum moment difference from the original mesh can be determined. The procedures are applied to some models and better results are obtained in comparison with some known algorithms.

5.1. Introduction

The volume visualization methods are divided into two categories: direct volume rendering and surface rendering methods [1]. For the later, the surface based volume visualization methods, a surface shell is extracted from the volume data. The classical approach to surface extraction is the Marching Cubes algorithm, proposed by Lorensen and Cline [2]. But one problem of this method is that it produces large number of triangles in the resulting mesh model which may largely exceeds the capacity of the graphics hardware real-time rendering [3]. One natural way to solve this problem consists of simplifying the mesh model by eliminating elements of polygons (vertices, edges, faces) for topologically-simple surface or by reducing the geometric and topological complexity for topologically-rich model [4], replacing the original object with fewer triangles while trying to keep its main characteristics.

The mesh simplification methods can be classified into four groups: sampling, adaptive subdivision, decimation and vertex merging.

- Sampling algorithms sample the geometry of the initial models, either with points upon the model's surface or voxels superimposed on the model in a 3D grid. They may have trouble achieving high fidelity since high frequency features are inherently difficult to sample accurately.
- Adaptive subdivision algorithms find a simple base mesh that can be recursively subdivided to more and more closely approximate the initial model.
- Decimation techniques iteratively remove vertices or faces from the mesh, re-triangulating the resulting hole after each step. These algorithms are relatively simple to code and very fast.
- Vertex merging schemes operate by collapsing two or more vertices of a triangulated model into a single vertex, which can in turn be merged with other vertices. Vertex merging is a fairly simple and easy-to-code mechanism, but algorithms use techniques of varying sophistication to determine which

vertices to merge in what order. Most view-dependent algorithms are based on vertex merging.

Among these methods, the vertex merging schemes which principally consist in iteratively removing edges or triangles from the mesh are very fast and relatively simple to program. According to the merged elements such as edges or triangles, the simplification method is called edge collapse or triangle removal, respectively.

Most of the vertex merging approaches is based on the following iterative framework:

- Assigning a cost to each valid operation which represents the amount of change introduced in the model.
- Applying the operation with minimum cost.
- Recalculating the costs of the operation belonging to the modified mesh portion. This iteration continues until the desired resolution is reached.

Mesh simplification based on vertex merging schemes introduces a metric between the original mesh and the approximated one. In the design of a mesh simplification algorithm, an important issue is the selection of an element to be deleted. Choosing optimal elements guarantees minimization of the metric between the original mesh and the final approximation. Oliver et al. [5] compared several metrics for mesh simplification and indicated the importance of the metric to the quality of approximations. These simplification metrics can be classified into two categories according to the geometry features they try to preserve: local features and global features. Most of the proposed metrics are based on local properties which guaranty preservation of local features [6-8]. Some authors started to imply the global features based metrics such as the area-based metric [9] and the volume-based metric [10]. The two metrics are both based on the object's global features, but their preserved characteristics are single and lack of enough information.

Moments and moment invariants contain more information about the object and they are widely used in object representation and recognition [11]. Since the moments of lower order (up to two) can be used to describe the shape of boundary segment, they have been successfully used to detect the image boundary [12, 13]. Shu et al. [14] proposed moment-based methods for polygonal approximation of digitized curves. Inspired by these research works, a surface moments-based metric and a volume moments-based metric are proposed with the purpose of generating low error approximations and being simple to implement. A uniform framework is applied for the simplification of all models, and only the investigated metrics are different. This framework is designed based on edge collapse algorithm, with the aim of comparing the different metrics independently of other aspects related to the simplification method.

5.2. Related previous works

As mentioned before, vertex merging based simplification process often takes an iterative method. The metric according to which we decide the simplification sequence is crucial during the iterative process. Some existing metrics are presented in this section.

Simplification metrics can be divided into two classes: local features-based metrics and global features-based metrics.

Within the class of local feature based metrics methods, one of the first vertex removal based simplification methods was proposed by Schroeder et al. [6]. The implemented simplification metric is the distance of the vertex to the average plane formed by its adjacent triangles. This metric is very simple to compute, but generates low-quality approximated models. Kim et al. [15] proposed a discrete curvature metric for simplification which has been demonstrated as not satisfactory because it always generates low-quality simplifications [5]. Vollmer et al. [7] used the standard deviation of a vertex set as a simplification metric. It has the effect of smoothing a triangle mesh during its simplification. Graland et al. [8] proposed a quadric error metrics (QEM) based algorithm. This algorithm made use of the quadric error metric to choose the edge to be simplified and the new vertex after contraction. It estimates the error introduced by a pair collapse operation as the distance from a vertex to a quadratic surface, represented as a symmetric matrix. The algorithm provides high-quality results because the quadric matrices are accumulated during the simplification process. Lindstrom et al. [16] added volume preservation and boundary preservation constrains to the quadratic objective functions when selecting the position of the new vertex. Hoppe [17] introduced an energy function to describe the complexity and fidelity of mesh and tracked simplification quality minimizing it. Hoppe's energy function requires many vertex distance evaluations so that it reduces the computational speed. Klein et al. [18] evaluated the Hausdorff distance between the original and simplified models, allowing precise error control. Hussain et al. [19] proposed a metric which is the summation of geometric change combined with vertex visual importance.

The above presented metrics are all designed according to the local feature preservation, but ignore the global geometry features, which can influence the approximation of simplified mesh. Wu et al. [20] introduced a global geometry features preservation method in the QEM based method. In this case, the global feature preservation is used as constraint within the simplification process and the metric itself is still according to the local features.

Methods directly based on global geometry features preservation are another efficient way to determine the approximated meshes. Among this kind of methods, Park et al. [9] proposed an area-based metric which compares the area difference of the original mesh S and its approximation S'. The cost function is defined as:

$$AD = \left| AREA(S) - AREA(S') \right| \tag{5.1}$$

Alliez et al. [10] proposed a volume-based metric, which is used to minimize the volume difference between the simplified mesh and the original mesh. The cost function is as follows:

$$VD = \left| VOLUME(S) - VOLUME(S') \right| \tag{5.2}$$

Inspired by these two metrics, we propose two moments-based simplification metrics with the purpose of improving the approximated results, which is described in detail in the following section.

5.3. Proposed moments-based metrics

In this section, the two moments-based metrics are presented. They are based on surface moments and volume moments respectively. The fast computation methods proposed by Tuzikov et al. [21] are applied to reduce the computational burden.

5.3.1. Surface moments-based metric

The surface moment of order $k_1+k_2+k_3$ of a 3D compact body P is defined as:

$$m_{k_1k_2k_3}S(P)=\int_{S(P)}x^{k_1}y^{k_2}z^{k_3}dS \tag{5.3}$$

where the integral is taken on the surface of P.

Notice that $m_{000}S(P)$ is the area of the model's surface which is used by Park et al. [9] as a simplification metric (Eq. (5.1)) in their method. Since the higher order moments contain more information about the object's surface, we will use the moments difference to measure the similarity between the original model surface S and the simplified model surface S'. Let $m_{k_1k_2k_3}S(P)$ and $m_{k_1k_2k_3}S'(P)$ be the surface moments defined by S and S', respectively. Then we define the following simplification cost function:

$$SMD=\left|\sum_{p=0}^{M}\sum_{q=0}^{p}\sum_{r=0}^{q}\left(m_{p-q,q-r,r}S(P)-m_{p-q,q-r,r}S'(P)\right)\right| \tag{5.4}$$

where M is the maximum order of moments that we will use.

Comparing Eq. (5.4) with Eq. (5.1), it can be seen that Eq. (5.1) corresponds to a special case of Eq. (5.4) (with $M = 0$). Therefore, we can expect to obtain better results using Eq. (5.4) with $M > 0$ contrasting with Eq. (5.1). However, the algorithm directly based on Eq. (5.4) could be time-consuming because the moment calculation by a straightforward method is very expensive. This problem can be solved using a simple and fast surface moment computing algorithm which was proposed by Tuzikov et al. [21]. A brief description of this algorithm is given below.

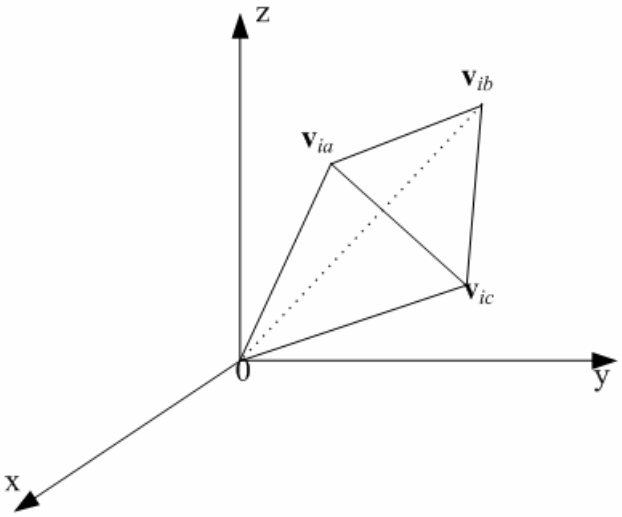


Figure 5.1: Tetrahedron T_i

Assume that the object is represented by a mesh of N triangles, each triangle defined by its vertices $\mathbf{v}_{ia}$, $\mathbf{v}_{ib}$, $\mathbf{v}_{ic}$, i = 1, 2, …, N. For each triangle we form a tetrahedron denoted by T_i defined by the three vertices and the coordinate origin (Fig. 5.1). Let $S_0(T_i)$ be the tetrahedron facet that is opposite to the coordinates origin, then Eq. (5.3) can be rewritten as:

$$m_{k_1k_2k_3}S(P)=\int_{S(P)}x^{k_1}y^{k_2}z^{k_3}dS=\sum_i m_{k_1k_2k_3}S_0(T_i) \tag{5.5}$$

Eq. (5.5) shows that we need only to calculate the surface moment $m_{k_1k_2k_3}S_0(T_i)$ of order $k_1+k_2+k_3$. To do this, let $T_i = T_i(\mathbf{v}_{ia}, \mathbf{v}_{ib}, \mathbf{v}_{ic})$ be the tetrahedron formed by the coordinate origin and the three vertices $\mathbf{v}_{ia} = (a_1, a_2, a_3)$, $\mathbf{v}_{ib} = (b_1, b_2, b_3)$, $\mathbf{v}_{ic} = (c_1, c_2, c_3)$ and the vertices $\mathbf{v}_{ia}$, $\mathbf{v}_{ib}$, $\mathbf{v}_{ic}$ are arranged in counter-clockwise order with respect to the outward normal of the surface $S_0(T_i(\mathbf{v}_{ia}, \mathbf{v}_{ib}, \mathbf{v}_{ic}))$.

Let us introduce some notations. Denote by $A=(A_{ij})$ the following matrix:

$$A=\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}.$$

Given integers k_1, k_2, k_3 denote by ζ a set of such 3×3 matrices (k_{ij}) with integer values k_{ij}, $0\le k_{ij}\le k_i$ such that $\sum_{j=1}^{3}k_{ij}=k_i$, i = 1, 2, 3. The following formula is derived for computing the surface moment $m_{k_1k_2k_3}S_0(T)$ of order $k=k_1+k_2+k_3$:

$$m_{k_1k_2k_3}S_0(T)=\frac{2Ar(S_0)k_1!k_2!k_3!}{(k+2)!}\times\sum_{(k_{ij})\in\zeta}\frac{\prod_{j=1}^{3}\left(\left(\sum_{i=1}^{3}k_{ij}\right)!\right)}{\prod_{i,j=1}^{3}(k_{ij}!)}\prod_{i,j=1}^{3}A_{ij}^{k_{ij}} \tag{5.6}$$

where $Ar(S_0)$ denotes the area of the face $S_0(T)$.

The fast surface moment computing algorithm allows us to develop a mesh simplification method based on higher order moments.

Using Eq. (5.5), Eq. (5.4) becomes:

$$SMD=\left|\sum_{p=0}^{M}\sum_{q=0}^{p}\sum_{r=0}^{q}\left(\sum_i m_{p-q,q-r,r}S_0(T_i)-\sum_i m_{p-q,q-r,r}S_0'(T_i)\right)\right| \tag{5.7}$$

Eq. (5.7) can be used to determine a sequence of approximation meshes with different number of triangles. Let F_N be the original mesh with N triangles. For a specified number n, the objective may be to find an approximated mesh F_n so that the moment differences of F_n from the initial mesh F_N is minimal. The candidate F_n is uncountable because the three-dimensional model is complicated. In order to simplify the optimization procedure, we take the iterative vertex merging methods (face removal or edge collapse) to determine a sequence of approximated meshes of the initial model. During the iterative vertex merging procedure, only the moments related to the modified triangles are changed for each step. So Eq. (5.7) can be simplified to a more efficient form. Here we take edge collapse procedure for illustration.

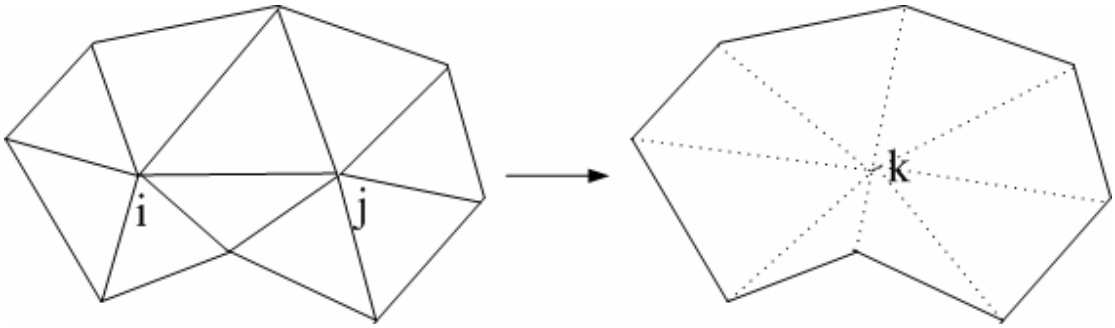


Figure 5.2: Process of edge collapse from e_{ij} to vertex k

As illustrated in Fig. 5.2, each operation will remove two triangles adjacent to the deleted edge e_{ij}. That is, we start with the initial mesh F_N to get the first simplified mesh F_{N-2}, then the simplified mesh F_{N-4} from F_{N-2}, and so on until finding F_n. The greedy strategy we use in the approximation process is that F_{N-2} is derived from F_N such that the surface moment difference between F_{N-2} and F_N is minimal (global minimum). To obtain the approximated mesh F_{N-2} from F_N, the kernel operation of the approximation process is to collapse an edge e_{ij} to a point k. Since all the other triangles have no change except the triangles adjacent to the two points i and j, which are the two vertices of the deleted edge. We can deduce from Eq. (5.7) that the moment difference of order up to M between F_{N-2} and F_N is as follows:

$$ESMD = \left| \sum_{p=0}^{M} \sum_{q=0}^{p} \sum_{r=0}^{q} \left(\sum_{v \in \left(T_i + T_j - T_{e_{ij}}\right)} m_{p-q,q-r,r} S_0(T_v) - \sum_{v \in T_k} m_{p-q,q-r,r} S_0(T_v) \right) \right| \qquad (5.8)$$

where T_v denotes the triangles adjacent to a vertex v and $T_{e_{ij}}$ denotes the two triangles adjacent to the edge e_{ij}. Eq. (5.8) shows that the computation of adjacent triangle moments is taken instead of the whole object triangle moments.

This principle can be extended to vertex or face removal methods by only taking their respective triangle changes caused by each operation into account.

5.3.2. Volume moments-based metric

The volume moment of order $k=k_1+k_2+k_3$ of a 3D compact body P is defined as:

$$m_{k_1 k_2 k_3} V(P) = \int_P x^{k_1} y^{k_2} z^{k_3} dxdydz \qquad (5.9)$$

where the integral is taken on the volume of P.

Notice that $m_{000}V(P)$ is the volume of the model P which is used by Alliez et al. [10] as a simplification metric (Eq. (5.2)) in their method. The simplification cost function by using the volume moments is similar to that of the surface moments. It is defined by

$$VMD = \left| \sum_{p=0}^{M} \sum_{q=0}^{p} \sum_{r=0}^{q} \left(m_{p-q,q-r,r} V(P) - m_{p-q,q-r,r} V'(P) \right) \right| \qquad (5.10)$$

where $m_{k_1 k_2 k_3} V(P)$ and $m_{k_1 k_2 k_3} V'(P)$ denote the $(k_1+k_2+k_3)$th-order of volume moments defined on the volume P and simplified mesh P', respectively. Comparing

Eq. (5.10) with Eq. (5.2), it can be seen that Eq. (5.2) corresponds to a special case of Eq. (5.10) (with $M = 0$).

Tuzikov et al. [21] also proposed a fast algorithm for computing the volume moments. The computation method of volume moments is similar to that of surface moments. Let $V(T_i)$ be the volume of the tetrahedron T_i (Fig. 5.1), similar to Eqs. (5.5), (5.6), the corresponding formulas for volume moments-based simplification are as follows:

$$m_{k_1k_2k_3}V(P) = \int_P x^{k_1} y^{k_2} z^{k_3} dxdydz = \sum_i m_{k_1k_2k_3}V(T_i) \tag{5.11}$$

$$m_{k_1k_2k_3}V(T) = \frac{|A|k_1!k_2!k_3!}{(k+3)!} \times \sum_{(k_{ij})\in\zeta} \frac{\prod_{j=1}^{3}\left(\left(\sum_{i=1}^{3} k_{ij}\right)!\right)}{\prod_{i,j=1}^{3}(k_{ij}!)} \prod_{i,j=1}^{3} A_{ij}^{k_{ij}} \tag{5.12}$$

where $|A|$ is the determinant of A.

Using Eq. (5.11), Eq. (5.10) becomes:

$$VMD = \left| \sum_{p=0}^{M}\sum_{q=0}^{p}\sum_{r=0}^{q} \left(\sum_i m_{p-q,q-r,r}V(T_i) - \sum_i m_{p-q,q-r,r}V'(T_i) \right) \right| \tag{5.13}$$

For iterative mesh simplification methods, only some local modification presents to each iteration. Similar to that of surface moments-based metric, we take edge collapse operation to illustrate the simplified form of Eq. (5.13) for a series of vertex merging methods. We can deduce from Eq. (5.13) that the moment difference of order up to M between F_N and F_{N-2} is as follows:

$$EVMD = \left| \sum_{p=0}^{M}\sum_{q=0}^{p}\sum_{r=0}^{q} \left(\sum_{v\in(T_i+T_j-T_{e_{ij}})} m_{p-q,q-r,r}V(T_v) - \sum_{v\in T_k} m_{p-q,q-r,r}V(T_v) \right) \right| \tag{5.14}$$

where *EVMD* denotes the global volume moment difference of an edge collapse operation, which is called the cost function. As mentioned previously, this metric can be easily extended to vertex or face removal operation.

5.4. A framework for mesh simplification metrics evaluation

In order to evaluate the proposed metrics, we follow the well-established mesh simplification framework described in [22]. This framework was implemented by Oliver et al. [5] to compare different simplification metrics, in which the authors suggest to employ the edge collapse operation (Fig. 5.2) for the simplification procedure because it preserves topology and it is easy to implement. In our opinion, it is a good choice because the approximation results mainly rely on the metrics, without the involvement of any other factors (like re-triangulation method choice).

The simplification procedure can be described as follows:

- Firstly, the costs of all candidate operations, in this case edge collapses that do not introduce degeneracy into the mesh, are calculated.
- The valid operations are inserted into a priority queue indexed by their costs.

- Thereafter, the iterative simplification procedure starts. The operation on the top of the queue is applied, and the costs of all operations dependent on the modified mesh portion are updated, possibly causing the candidates to change positions in the priority queue.
- This procedure continues until a desired number of mesh faces is removed from the model.

The metrics used to assign a cost to the possible operations are discussed in the following section. Instead of half-edge collapses implemented in [5], we performed a simple edge collapses procedure by adding a third choice for the position of *k*, which is the middle point of the collapsed edge.

5.5. Metrics evaluation method

The two moments-based metrics were evaluated quantitatively and estimated by measuring the global simplification error assessment. Since the root mean square (*RMS*) error measures the global average error between the model and its approximation, it is commonly used as the efficiency measurement in the mesh simplification algorithms. We used the error detection tool Metro [23] to calculate the *RMS* error between one model and its approximation. The *RMS* error from S' to S is defined as:

$$d_{rms}(S,S') = \sqrt{\frac{1}{AREA(S)}\int_S d(p,S')^2\, ds} \tag{5.15}$$

Besides the error detection method, the moments themselves can also be used as evaluation tools. The moments difference between the approximation and the original model is an efficient measure tool to evaluate the moments-preservation ability of the simplification methods. We have done some experiments to demonstrate this evaluation method in section 5.6.

5.6. Experiments and discussions

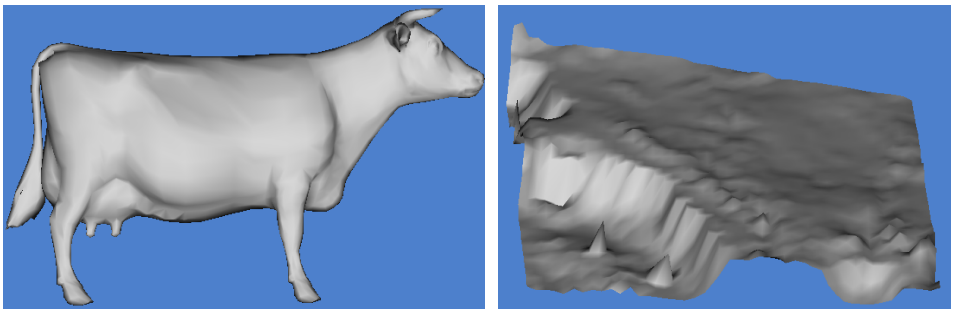
The experiments were performed on a PC Pentium 4 2.66GHZ CPU with 512MB RAM, running on Windows XP operating system. Visual C++ and OpenGL were taken as development tools.

5.6.1. Experimental models

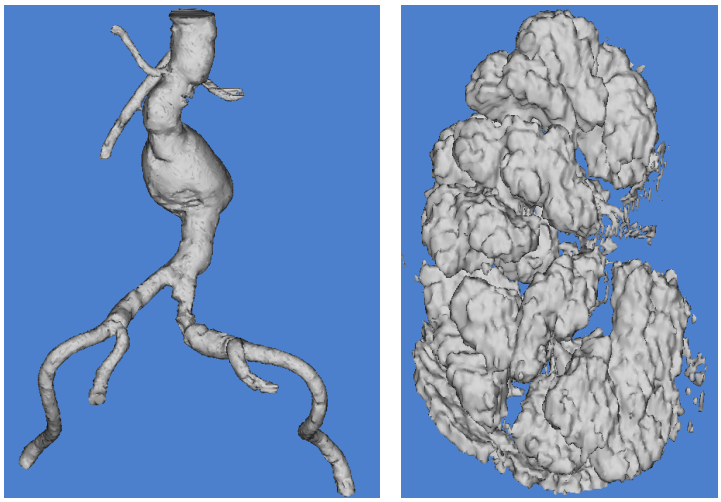
Four models are used to demonstrate the performance of the algorithm. They are: the Cow model, the North America model, the Vessel model and the Renal medulla model. Table 5.1 illustrates the detail information of these models. The original models are shown in Fig. 5.3.

Table 5.1: Information about the models in experiments

Models	Original vertices	Original triangles
Cow model	2904	5804
North America model	2025	3872
Vessel model	45029	90050
Renal medulla model	90438	180140



(a) Cow model (b) North America model



(c) Vessel model (d) Renal medulla model

Figure 5.3: Models for experiment

5.6.2. Implemental details

Based on the edge collapse method, the greedy strategy for determining the approximated meshes can be summarized as follows. In each step of the approximation process, i.e., when we pass F_N and F_{N-2}, the certain edge among all the edges of F_N is removed if it gives the minimum moment difference value. The corresponding algorithm is as follows:

Input: The original mesh F_N containing N triangles, the maximum order of moments M and the number of triangles n required.

Step 1: Compute every triangle's moments using Eq. (5.6) or Eq. (5.12) and then compute the moment difference (ESMD or EVMD) for each edge collapse operation using Eq. (5.8) or Eq. (5.14). The moment differences are stored as a contraction cost for every edge.

Step 2: Sort the costs in an increasing order.

Step 3: Select the top edge in the queue, and check if it can be contracted. If not, remove it from the queue and return to the beginning of step 3. If yes, contract this edge and recalculate the information of the affected triangles and edges.

Step 4: Update the position of the affected edges in the cost queue.

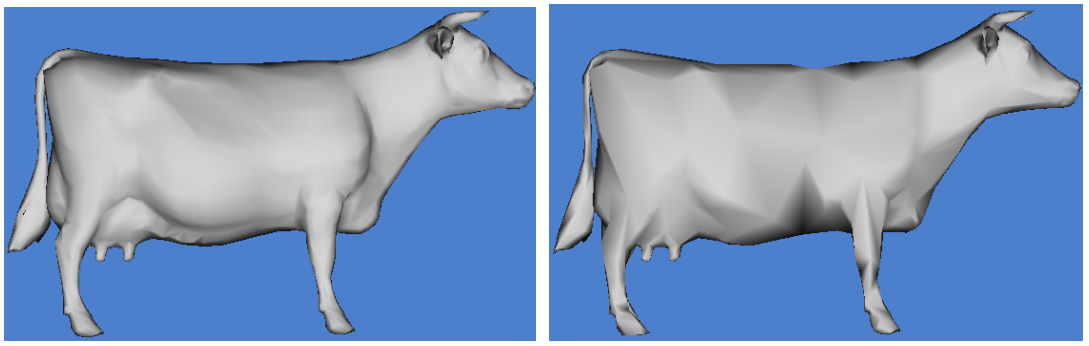
Step 5: Repeat step 3 and step 4 until the required triangle number of the approximation is reached.

Each time when one edge is collapsed, two triangles and one edge will be deleted. In order to perform the operation, we must choose a position for k. A simple scheme

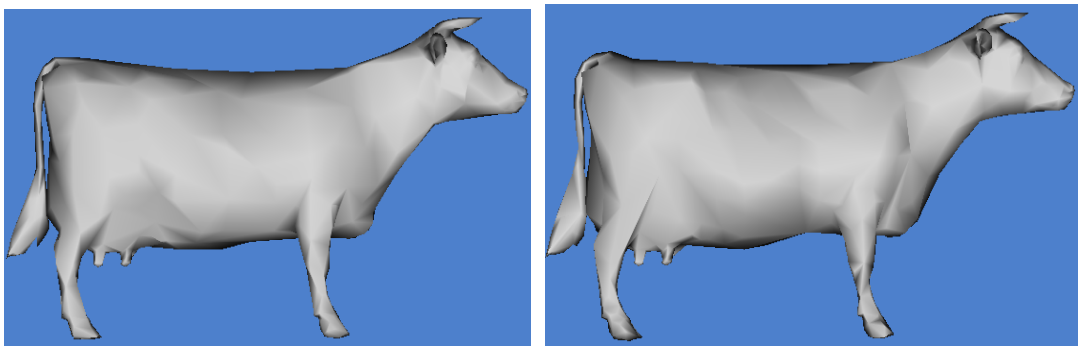
is implemented, which is to choose v_i, v_j or $(v_i + v_j)/2$ depending on the lowest value of simplification cost that produced by the new position. In order to preserve the vision characteristic, if the largest normal direction change of one edge's adjacent triangles is greater than a certain threshold (here we choose $\pi/4$) after edge contraction, we keep the edge.

5.6.3. Experimental results

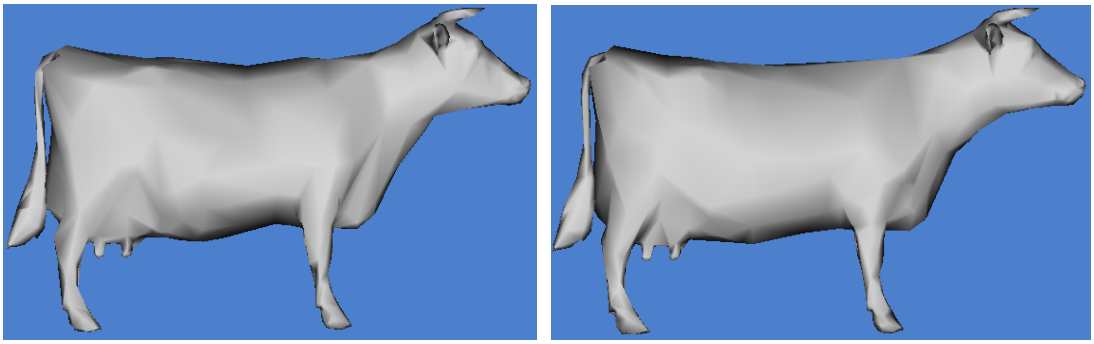
For the performance testing, we use the cow model and the North America model because they contain relatively fewer triangles and can better illustrate the performance of the simplification procedure. But the algorithm is also implemented on the vessel and the renal medulla models and the results are illustrated at the end of the section.



(a) Original Model (b) M=0



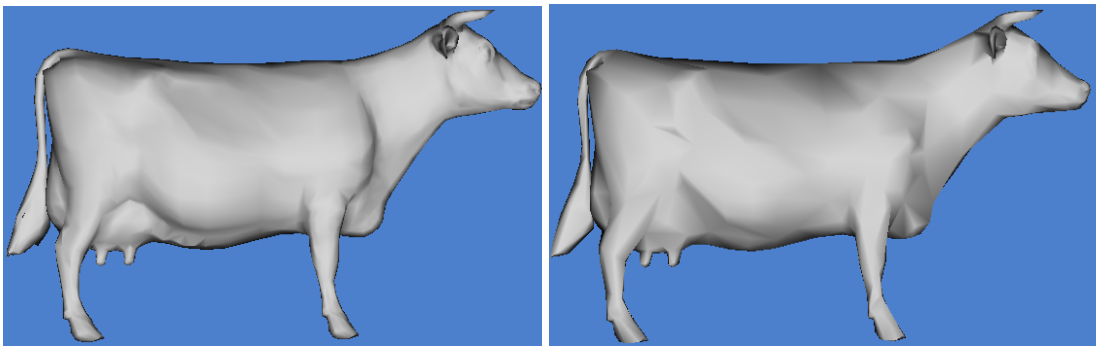
(c) M =1 (d) M =2



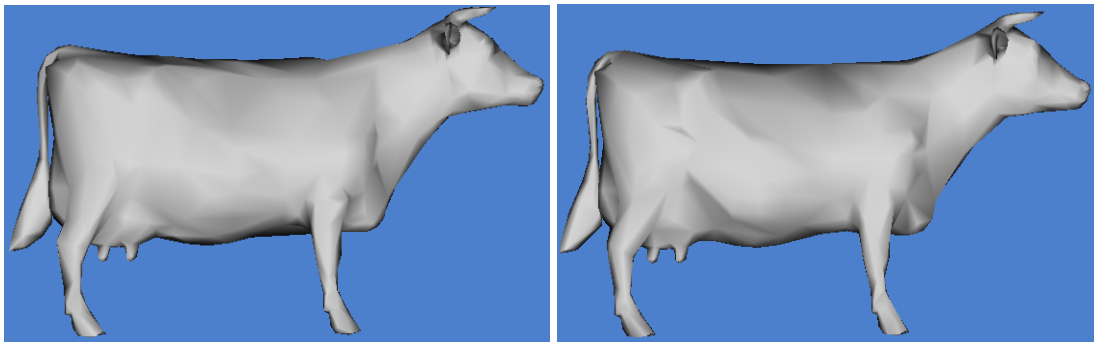
(e) M =3 (f) M =4

Figure 5.4: 70% simplification (possessing 30% of the original faces) for cow model using surface moments-based metric up to different maximum order M.

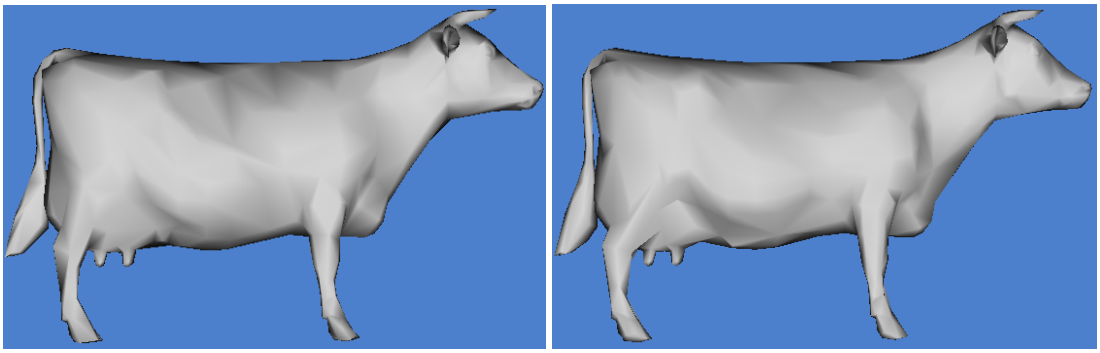
Fig 5.4 and 5.5 present the qualitative results of a 70% model simplification rate of the cow model (from 5804 triangles to 1704 triangles) using respectively surface and volume moments-based metrics for different maximum moment order values M. In all the cases we can see that they still contain the major topological characteristic of the initial model. From Fig. 5.4, we can see that the choice of $M > 0$ gets qualitatively better results than that of $M = 0$. When $M = 0$, the surface moments-based metric becomes the area-based metric of Eq. (5.1). Fig 5.5 shows that the increase of the moment order for volume-based method slightly improves the approximated results, but it performs better than the surface moments-based simplification.



(a) Original Model (b) M=0



(c) M =1 (d) M =2



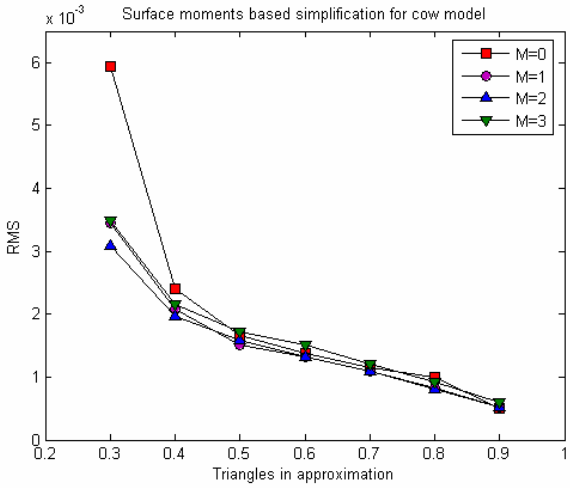
(e) M =3 (f) M =4

Figure 5.5: 70% simplification (possessing 30% of the original faces) for cow model using volume moments-based metric up to different maximum order M.

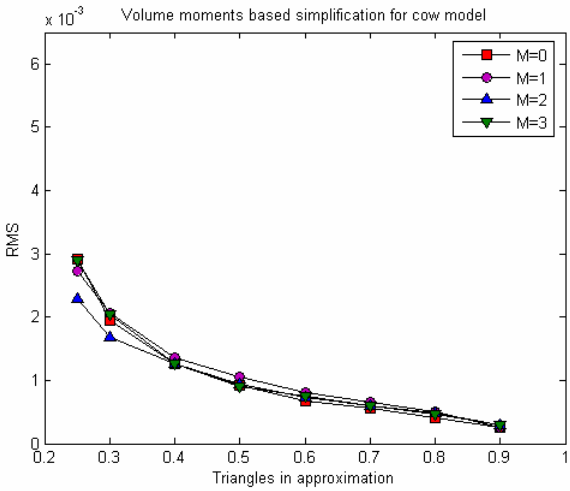
Fig. 5.6 and Fig. 5.7 show the simplification errors of the two moments-based metrics for different values of M in terms of the simplification rate. The results

presented in Fig 5.6 (a) and (b) confirm quantitatively the visual effect of Fig. 5.4 and Fig. 5.5, respectively. For the two moments-based metrics, $M = 0$ represents area-based metric and volume-based metric respectively. When the required facet rate in the approximation is low, it seems that $M > 0$ can get a lower simplification error than $M = 0$. That is to say, the moments-based metrics can get a better result in the simplification error sense.

In order to compare the two moments-based metrics, a graph of their simplification errors is drawn in Fig. 5.8. For each metric, we choose the lowest simplification error for M varying from 0 to 3. It seems that the volume moments-based metric provides lower simplification error.

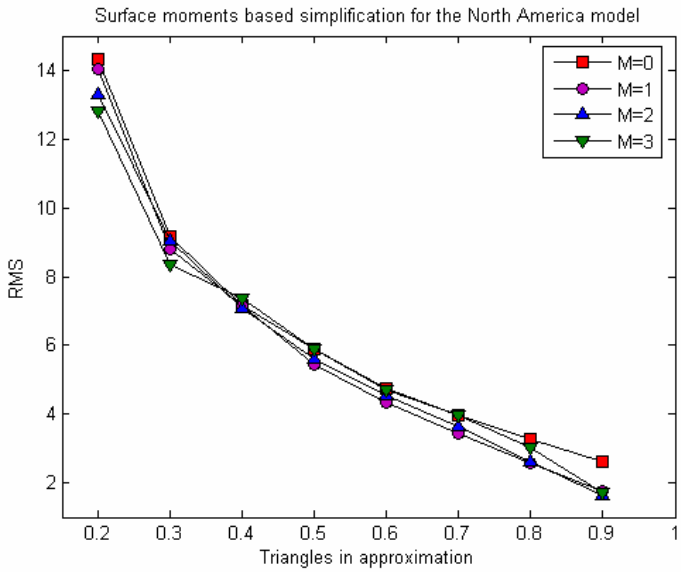


(a) Surface moments

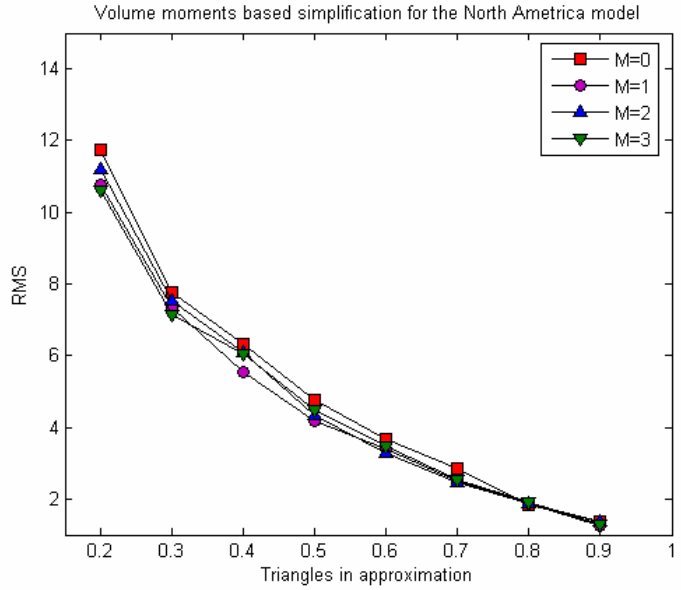


(b) Volume moments

Figure 5.6: Influence of the moment order on the *RMS* error between the cow model and its approximation for different simplification rate: (a) Surface moments (*SM*)-based method. (b) Volume moments (*VM*)-based method.

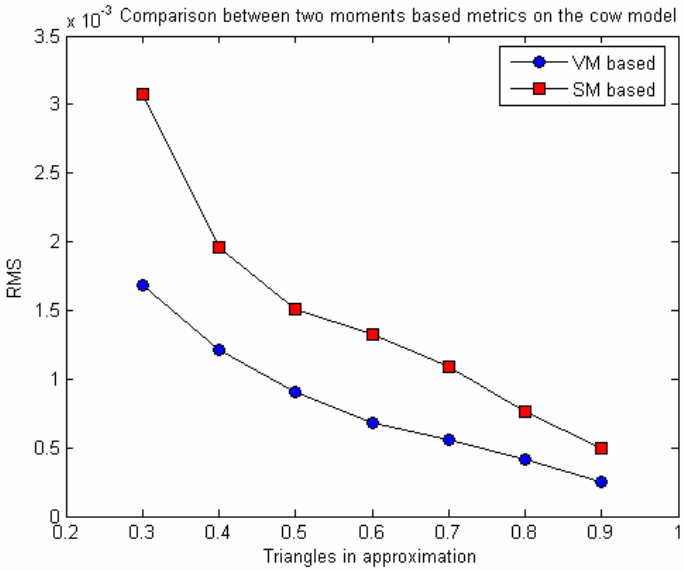


(a) Surface moments

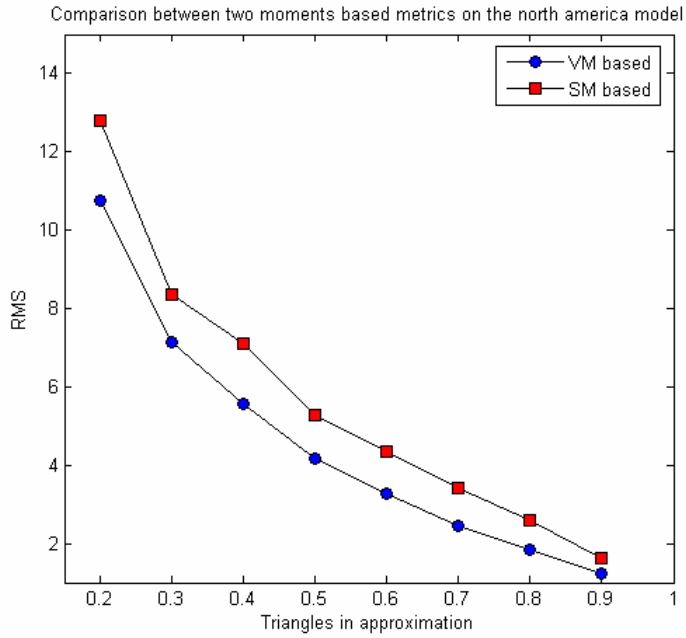


(b) Volume moments

Figure 5.7: Influence of the moment order on the *RMS* error between the North America model and its approximation for different simplification rate: (a) Surface moments (*SM*)-based method. (b) Volume moments (*VM*)-based method.



(a) the cow model



(b) the North America model

Figure 5.8: Comparison of *VM*-based method and *SM*-based method: (a) the cow model. (b) the North America model

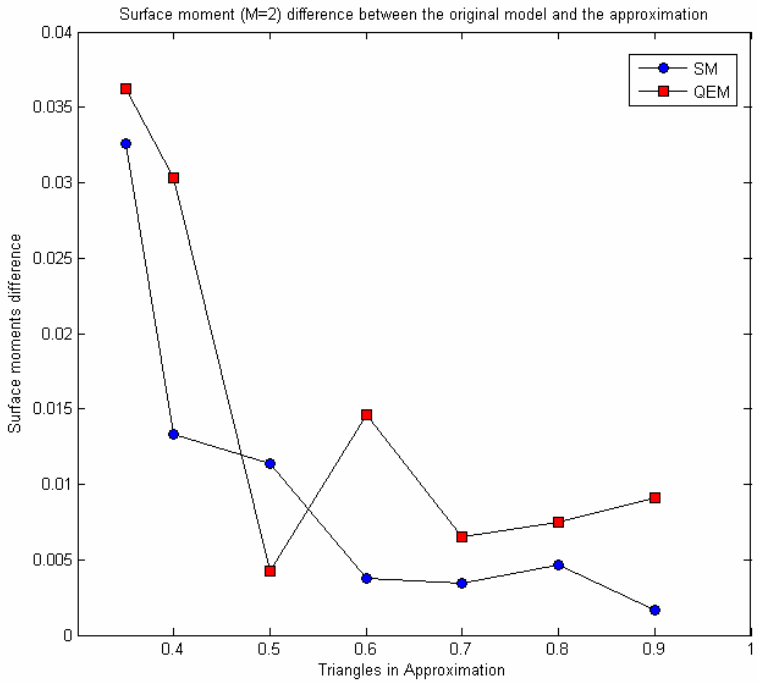


Figure 5.9: Comparison of SM differences between SM-based method and QEM

The aim of the proposed metrics is to preserve the moment features of the original models. So in order to demonstrate the preservation ability of the two metrics, we compute the moment difference between the original model and the approximation, using our metrics and the quadric error metric (QEM) respectively. The results are illustrated in Fig. 5.9 and Fig. 5.10, for SM-based metric and VM-based metric respectively. Here we choose $M = 2$ for illustration. The results show that our methods have a better ability to preserve the moment features. These results also demonstrate that our global moments-based metrics can also be used to evaluate the several mesh simplification methods.

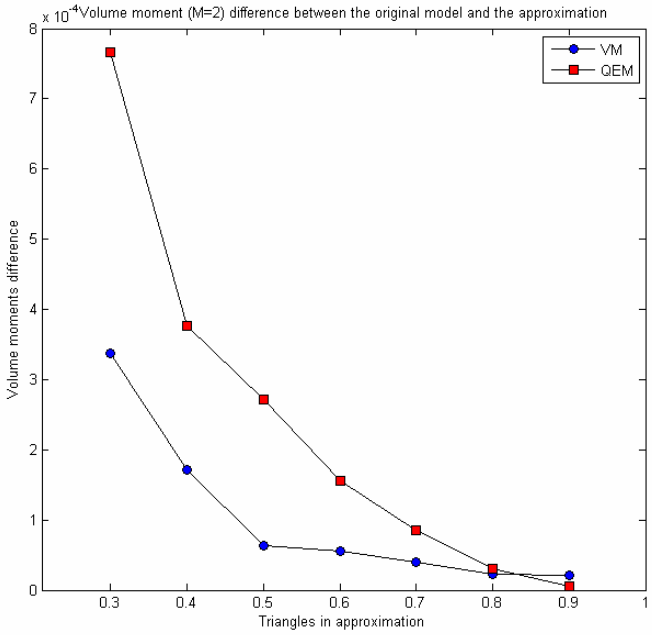
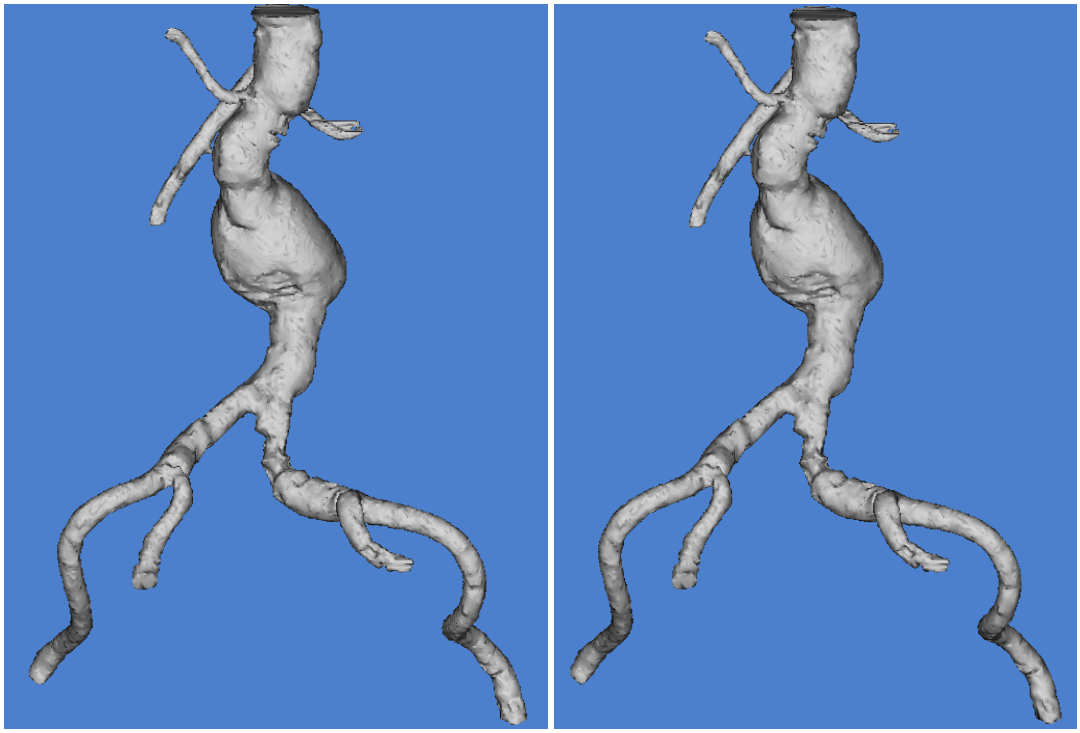


Figure 5.10: Comparison of VM differences between VM-based method and QEM

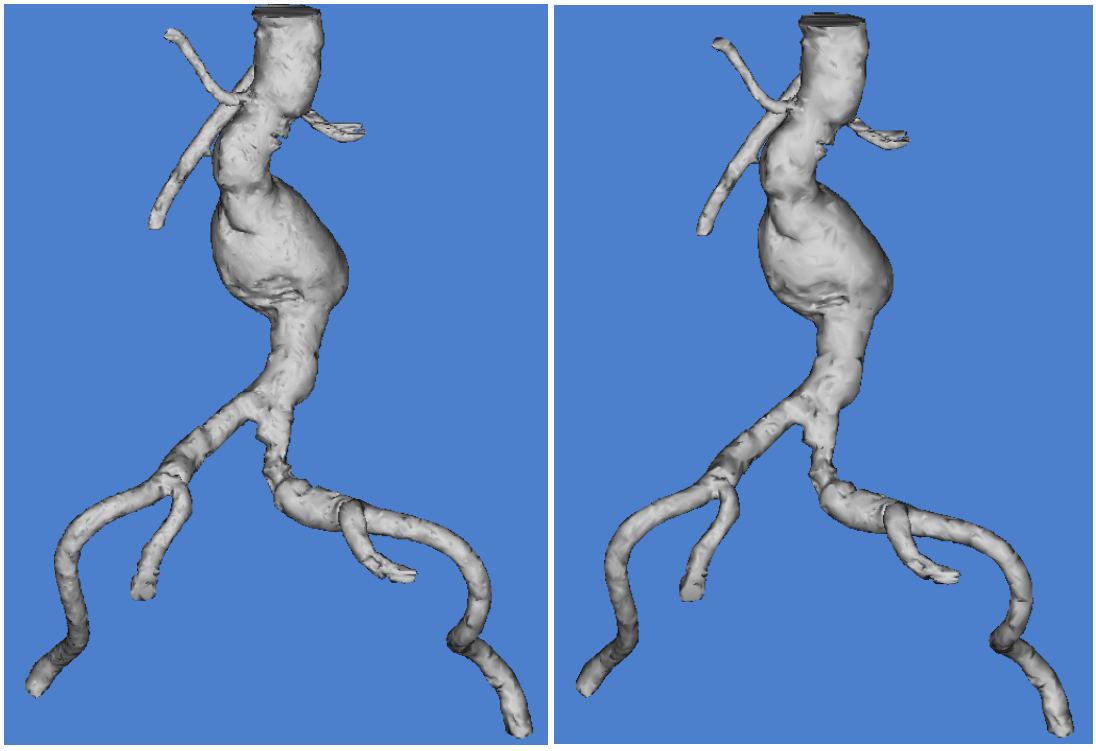
From the above discussion, we can see that VM-based metric can get a better performance in both appearance and simplification error senses. We implemented it to the extracted vessel model and the renal medulla model. Here we choose M=2 for illustration. The experimental results are illustrated in Fig. 5.11 and Fig. 5.12 respectively. It can be seen that the appearance and the structures are kept very well while the number of triangles is largely deduced, as illustrated in Table 5.2.

Table 5.2: Number variation of vertex and triangle in the process of simplification of vessel model and renal medulla model

Vessel model	Original	70%	50%	20%
Vertices	45209	31521	22516	9009
Triangles	90050	63034	45024	18010
Renal medulla model	Original	70%	50%	30%
Vertices	90438	63416	45403	25389
Triangles	180140	126096	90070	54042

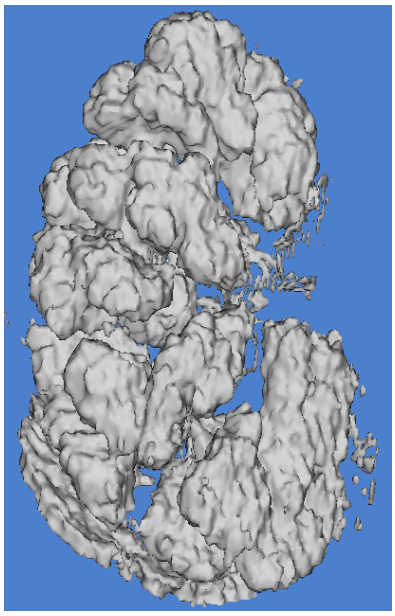


(a) Original model (90050 triangles) (b) 70% (63034 triangles)

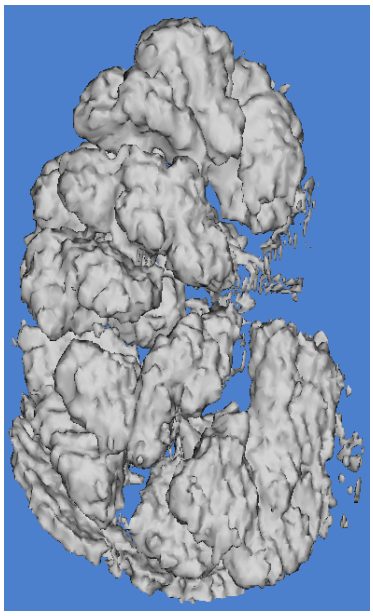


(c) 50% (45204 triangles) (d) 20% (18010 triangles)

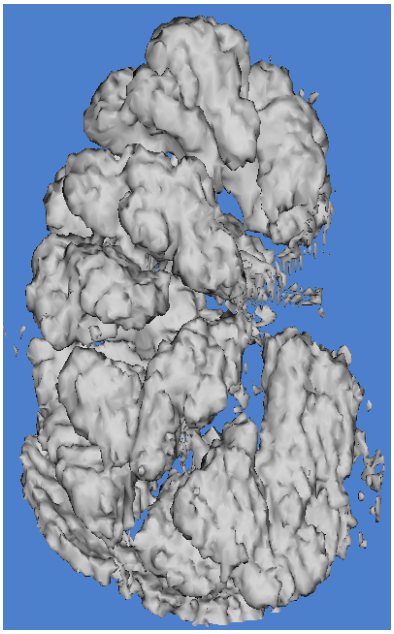
Figure 5.11: Mesh simplification results on the vessel model using VM-based metric (M=2). (a) The original model, (b) possession 70% of the original model, (c) possession 50%, (d) possession 20%.



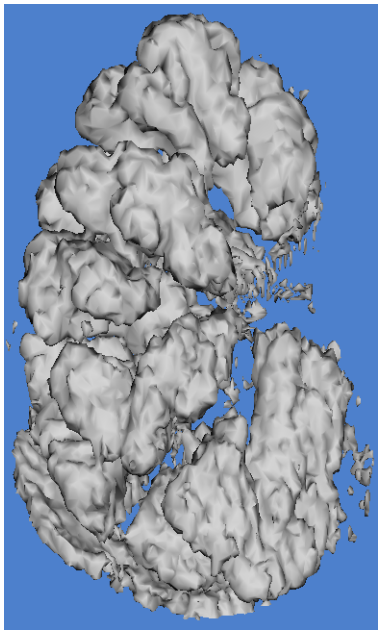
(a) Original model (180140 triangles)



(b) 70% (126096 triangles)



(c) 50% (90070 triangles)



(d) 30% (54042 triangles)

Figure 5.12: Mesh simplification results on the renal medulla model using VM-based metric (M=2). (a) The original model, (b) possession 70% of the original model, (c) possession 50%, (d) possession 30%.

5.7. Conclusions

In this chapter, we proposed two new metrics for mesh simplification, with the aim of preserving the original model's global features in the approximation. They are designed based on the surface moments and volume moments by extending the two global metrics, which are area based and volume-based metrics, respectively. The experimental results demonstrated that the extended metrics can get better result since they contain more information about the model. From the experimental analysis, we

can get the conclusion that volume-based metrics can achieve a better performance than the surface one. We also demonstrate that the moments based metrics can also be used as an evaluation metric for the comparison of mesh simplification algorithms. Finally, we implemented the proposed algorithm in our medical visualization framework.

References

[1] T. T. Elvins, "A survey of algorithms for volume visualization," presented at SIGGRAPH '92, Chicago, IL, 1992, pp. 194-201.

[2] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface reconstruction algorithm," Computer Graphics, vol. 21, pp. 163-169, 1987.

[3] K. Brodlie and J. Wood, "Recent advances in volume visualization," Computer Graphics Forum, vol. 20, pp. 125-148, 2001.

[4] C. Andujar, P. Brunet, and D. Ayala, "Topology-reducing surface simplification using a discrete solid representation.," ACM Trans. on Graphics, vol. 21, pp. 88-105, 2002.

[5] M. Oliver and P. Hélio, "A comparative evaluation of metrics for fast mesh simplification," Computer Graphics Forum, vol. 25, pp. 197-210, 2006.

[6] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," presented at SIGGRAPH '92, Chicago, IL, 1992, pp. 65-70.

[7] J. Vollmer, R. Mencl, and H. Muller, "Improved Laplacial smoothing of noisy surface meshes," Computer Graphics Forum, vol. 18, pp. 131-138, 1999.

[8] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," presented at SIGGRAPH '97, Los Angeles, CA, 1997, pp. 209-216.

[9] I. Park, S. Shirani, and D. W. Capson, "Mesh simplification using an area-based distortion measure," Journal of Mathematical Modeling and Algorithms, vol. 5, pp. 309-329, 2006.

[10] P. Alliez, N. Laurent, H. Sanson, and F. Schmitt, "Mesh approximation using a volume-based metric," presented at the 7th Pacific Conference on Computer Graphics and Applications Seoul, South Korea, 1999, pp. 292-301.

[11] R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," CVGIP: Graphical Models and Image Processing vol. 54, pp. 438-460, 1992.

[12] S. Ghosal and R. Mehrotra, "Orthogonal moment operators for subpixel edge detection," Pattern Recognition, vol. 26, pp. 295-306, 1993.

[13] L. M. Luo, C. Hamitouche, J. L. Dillenseger, and J. L. Coatrieux, "A moment based three dimensional edge operator," IEEE Trans Biomed Eng, vol. 40, pp. 693-703, 1993.

[14] H. Z. Shu, L. M. Luo, Z. D. Zhou, and X. D. Bao, "Moment-based methods for polygonal approximation of digitized curves," Pattern Recognition, vol. 35, pp. 421-434, 2002.

[15] S. J. Kim, W. K. Jeong, and C. H. Kim, "LOD generation with discrete curvature error metric," presented at 2nd Korean Israel Bi-National Conference on Geometrical Modeling and Computer Graphics in the World Wide Web Era, Seoul, Korea, 1999, pp. 97-104.

[16] P. Lindstrom and G. Turk, "Fast and memory efficient polygonal simplification," presented at 9th IEEE Visualization 98, Research Triangle Park, NC, 1998, pp. 279-286.

[17] H. Hoppe, "Progressive meshes," presented at SIGGRAPH' 96, New Orleans, LA, 1996, pp. 99-108.

[18] R. Klein, G. Liebich, and W. Strasser, "Mesh reduction with error control," presented at IEEE Visualization 1996 Burlingame, CA, 1996, pp. 311-318.

[19] M. Hussain, Y. Okada, and K. Niijima, "Efficient and feature-preserving triangular mesh decimation," Journal of WSCG, vol. 12, pp. 167-174, 2004.

[20] Y. Wu, Y. He, and H. Cai, "QEM-based mesh simplification with global geometry features preserved," presented at 2nd international conference on Computer Graphics and interactive techniques in Australasia and South East Asia, Singapore, 2004, pp. 50-57.

[21] A. V. Tuzikov, S. A. Sheynin, and P. V. Vasiliev, "Computation of volume and surface body moments," Pattern Recognition, vol. 36, pp. 2521-2529, 2003.

[22] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner, Level of detail for 3D graphics. San Francisco: Morgan Kaufmann, 2003.

[23] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: measuring error on simplified surfaces," Computer Graphics Forum, vol. 17, pp. 167-174, 1998.

Chapter 6: Conclusions and future works

This chapter summarizes the conclusions of our research and outlines several ideas for the future work based on our current results.

6.1. Conclusions

Scientific visualization is currently a very active and vital area of research especially in medical area. Many researchers focus on the medical visualization problems. This dissertation takes the preoperative kidney planning system as a specific case to do some research about medical volume visualization techniques. The input data of the system are three to four time-spaced 3D acquisitions, which give relatively complementary information about the kidney anatomy. Our work followed the essential stages of the design of a general visualization tool: *registration*, *segmentation* (*classification*), *visualization* (*graphical representation*).

We did some research aiming at solving some of the problems that appeared during the visualization progress. Our work can be summarized as follows:

1) The classical urinary imaging system is the Spiral CT Urography, which gives three to four time-spaced acquisitions at different injection diffusion phases respectively. These acquisitions give relatively complementary information about the kidney anatomy. It is useful for the surgeon to integrate this information within a unique spatial volume. The first step in this integration process is to bring the different acquisitions into spatial alignment, which is called registration. To achieve this goal, a local MI maximization registration method is proposed. The kidneys are first extracted from the abdomen volumes and then the registration is performed between the extracted kidneys instead of the whole abdomen volumes. For the choice of registration metric, we implement an optimization independent protocol to evaluate several registration metrics and finally choose MI based metric for our practical situation. Optimization method is another important issue for registration methods. We find out that if the initial parameters are well chosen, the importance of optimization method can be reduced. We apply a geometric moments based registration technique to initialize the parameters and choose the relatively fast optimization method: downhill simplex method. Some implemental details, such as the choice of histogram resolution and interpolation method are also discussed. Experiments are performed on both synthetic and real data. The experimental results demonstrate the effectiveness of the kidney-centered registration method.

2) After registration of these acquisitions, we have a vectorial volume, which contains complete anatomical information. In order to outline the anatomical structures, multi-dimensional classification is necessary for analyzing this vectorial volume. Because of the partial volume effect (PVE), one voxel contains more than one material so that probability distributions are assigned to the different material types within this vectorial volume instead of a definite material label. Gaussian mixture model is often used in probability classification problems to model such distributions, but it relies only on the intensity distributions, which will lead a misclassification on the PVE boundaries and inhomogeneous regions with noises. In order to solve this

problem, a neighborhood weighted Gaussian mixture model is proposed. The model is that the voxels' intensity vectors follow the Gaussian mixture distribution and the classes distributions on each voxel are affected by its neighbors' class probability distributions so that a neighborhood weight is used to describe this property. Expectation Maximization algorithm is used as an optimization method to get the maximum likelihood estimation parameters of the neighborhood weighted Gaussian mixture model. The experiments demonstrate that the proposed method can get a better classification result and is less affected by the noise.

3) For the visualization of vectorial volume, most of the existing methods mix the component volumes at one certain step of the general scalar volume rendering pipeline. The problem of these methods is that they consider the component volumes individually instead of an integrated vectorial volume. We propose to analyze the vectorial volume directly instead of the individual component volumes. The first intermixing step can be achieved by the vectorial volume classification method we discussed before. After classification, several rendering techniques for scalar volume visualization can be adapted to our situation. Both of the surface rendering and volume rendering techniques are adapted to our practical situation. The comparison of these methods are given and discussed. Comparing to other methods, our method is an acquisition level intermixing method, which is the earliest intermixing stage during the rendering pipeline.

4) For surface-based volume visualization methods, triangle meshes are often used to represent the object surfaces. But the total number of triangles used to represent the object often largely exceeds the capacity of real-time rendering of graphics hardware. One nature way to solve it is to simplify the mesh models, replacing the original object with respectively fewer faces while trying to keep its main characteristics. The simplification metric is a key issue of a simplification algorithm. Two new simplification metrics based on surface moments and volume moments are proposed, which take the difference between the moments defined by the original mesh and those of the simplified mesh as the objective function. Edge collapse scheme is implemented to the mesh simplification procedure. For a given maximum moment order and the number of triangles required, the optimal mesh with a minimum moment difference from the original mesh can be determined. The procedures are applied to some models and better results are obtained in comparison with some known algorithms.

The research in this thesis is about a visualization tool for the preoperative kidney planning system with CT uroscans. But the proposed algorithms and techniques are not limited to this special application case. They can be adapted to other organs, even other non-medical application areas.

6.2. Future work

Medical volume visualization is a vital research topic for the development of medical applications. It involves research in computer graphics, image processing,

high performance computing and other areas. This dissertation only concerns a tiny part of the area. The future work can be done along the following paths:

1) For the conceptualization of the scene, automatic and more precise volume classification (segmentation) can be a research direction. Current classification method only gives the rough material distribution about the regions, with the loss of the volume detail information. A coarse-to-fine classification method would probably solve this problem.

2) For the rendering techniques, the future work can focus on containing more information in the final image with a better image quality. Achieving a faster rendering speed without any image quality loss is a continuous research topic, which can still be a research direction in the future.

3) For our specific application case: kidney preoperative planning system, there are still many tasks to be done: automatically delineating the renal carcinoma, specifying the renal arterial, venous and collecting system anatomies (by using for example a coarse-to-fine classification), fast rendering techniques for visualization etc.

Résumé

Ce mémoire de Thèse se focalise sur certains des problèmes non résolus en visualisation scientifique. Plus particulièrement nous avons pris une problématique médicale bien spécifique, la chirurgie conservatrice des tumeurs rénales, comme cadre applicatif pour l'élaboration de nouvelles solutions incluant des techniques de recalage de données, de segmentation et de visualisation 3D.

L'uroscan fournit 3 à 4 volumes présentant une information complémentaire sur l'anatomie rénale. La première étape consiste à mettre en correspondance ces différents volumes par une technique de recalage rigide du volume rénal basée sur la maximisation locale de l'information mutuelle.

L'idée principale de ce mémoire de Thèse est de proposer une visualisation de l'anatomie rénale directement à partir de ces données fusionnées. Pour cela, une technique de classification statistique des données basée sur une modélisation de la distribution des valeurs par un mélange de Gaussiennes incluant une information spatiale a été développée. Différentes techniques de visualisation 3D ont ensuite été adaptées à la représentation de cette information et comparées entre-elles.

Les techniques de représentation de surfaces peuvent être accélérées par des procédures de simplifications de maillages. Dans ce cadre, nous avons proposé deux métriques de description de la surface basées sur les moments géométriques et pouvant être incluses dans une telle procédure.

Ces différentes solutions, même si elles ont été développées dans le cadre de la représentation des structures anatomiques rénale, sont suffisamment génériques pour être utilisées ou adaptées à d'autres organes ou à d'autres applications médicales.

Abstract

This dissertation focuses on the main elements of a scientific visualization tool and takes a kidney preoperative information review system as a special application example to introduce the corresponding algorithms. Our research work followed the essential stages of the design of the kidney visualization system: registration, segmentation and visualization.

The CT uroscan consists of three to four time spaced 3D acquisitions, which give complementary information about the kidney anatomy. In order to bring these acquisitions into spatial alignment, a kidney centered registration method which is realized by local mutual information maximization is proposed. In order to illustrate the information contained in the spatial aligned volume, an acquisition level intermixing method is proposed, which intermix the several component volumes at the earliest stage. The first step for the acquisition level intermixing is a vectorial volume classification. We proposed a neighborhood weighted Gaussian mixture model, which involves the spatial information into the classification process. Then, several possible rendering techniques that can be adapted to this situation are presented and compared. For surface based volume visualization methods, mesh simplification is a usual way to improve rendering speed. The simplification metric is a key issue of a simplification algorithm. Two new mesh simplification metrics are proposed. They are based on surface moments and volume moments respectively.

Although these algorithms are introduced in the framework of the kidney visualization system, they are not limited to this system and can also be adapted to other applications.