

**ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC**

**MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**

**COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN TECHNOLOGIE DES SYSTÈMES
M.Ing.**

**PAR
MARC FOURNIER**

**FUSION DE DONNÉES 3D PROVENANT
D'UN PROFILOMÈTRE TENU À LA MAIN**

MONTRÉAL, MARS 2002

© droits réservés de Marc Fournier

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

- M. Richard Lepage, directeur de mémoire
Département de génie de la production automatisée
École de technologie supérieure
- M. Éric Harvey, codirecteur
Secteur des systèmes optiques et numériques
INO
- M. Mohamed Cheriet, président du jury
Département de génie de la production automatisée
École de technologie supérieure
- M. Tanneguy Redarce, professeur
Institut national des sciences appliquées de Lyon

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 14 MARS 2002

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FUSION DE DONNÉES 3D PROVENANT D'UN PROFILOMÈTRE TENU À LA MAIN

Marc Fournier

SOMMAIRE

Ce mémoire a été réalisé en collaboration avec l'INO, un centre de recherche en optique situé à Sainte-Foy, Québec. L'INO a développé un appareil portatif de numérisation trois dimensions (3D) par triangulation active, opérable par balayage à la main. Le système se compose d'une caméra CCD, d'une source de lumière structurée (ligne laser) et d'un système de positionnement 3D à six degrés de liberté servant à la mise en correspondance des profils 3D. Il fonctionne par le balayage à la main de la tête optique sur un objet à l'intérieur d'un volume défini par le système de positionnement 3D qui est basé sur une technologie ultrasonique. Les données fournies par le capteur sont sous forme de nuages de points 3D qui sont préalablement traités pour obtenir plusieurs surfaces dans l'espace 3D qui correspondent aux balayages effectués. Plusieurs balayages sont nécessaires pour numériser un objet en entier.

Chacune des surfaces partielles de l'objet représentée par un modèle polygonal triangulaire explicite est transformée dans un domaine de représentation implicite appelé la fonction de champ scalaire 3D. Cette fonction de champ est une grille 3D définie autour de la surface et à chaque élément de la grille une information de distance à la surface est conservée. Toutes les fonctions de champ individuelles sont intégrées en une seule fonction de champ globale qui représente la fusion des surfaces en une seule représentation implicite de l'objet. Les zones de redondance dans les surfaces individuelles sont ainsi éliminées. Une triangulation explicite est appliquée sur la fonction de champ globale pour revenir à une surface polygonale unique dans l'espace 3D qui représente l'ensemble de l'objet numérisé.

Le traitement est optimisé pour minimiser la mémoire nécessaire et les temps de calcul non négligeables de cette application. Des méthodes efficaces de recherche, de tri et de stockage des données ont été élaborées. Les résultats obtenus sont en format natif et un traitement ultérieur est réalisé pour les convertir dans un format standard supporté par les principaux logiciels utilisés en conception assistée par ordinateur. Ces résultats sont très satisfaisants pour la plupart des applications et les erreurs de surfaces introduites par le traitement sont inférieures à la précision du capteur utilisé. Dans certains cas les résultats présentent localement de légères imperfections quantifiables de surface.

REMERCIEMENTS

Je tiens tout d'abord à remercier M. Richard Lepage mon directeur de mémoire pour ses précieux conseils particulièrement lors de la rédaction de ce mémoire ainsi que M. Éric Harvey mon codirecteur à l'INO pour ses idées directrices lors du développement et de la gestion du projet dont ce mémoire fait partie intégrale.

Je remercie également la direction de l'École de technologie supérieure et celle de l'INO pour avoir rendu possible cette collaboration à la réalisation de ce mémoire et pour m'avoir fait confiance en me confiant ce projet.

Je remercie tout particulièrement M. Jean-François Lavoie, chercheur à l'INO, pour ses idées, ses conseils et son esprit critique à l'implémentation de ce projet. Sans lui la réalisation concrète de ce mémoire aurait été beaucoup plus longue et difficile.

Finalement je remercie les membres de ma famille ainsi que mes amis pour leur soutien et leurs encouragements tout au long de la réalisation de ce mémoire. Merci à Fanie et à Maïté aussi pour leur patience et leur compréhension lors des longues heures que j'ai passé à travailler sur ce mémoire.

TABLE DES MATIÈRES

	Page
SOMMAIRE	i
REMERCIEMENTS	ii
TABLE DES MATIÈRES.....	iii
LISTE DES FIGURES	v
INTRODUCTION.....	1
CHAPITRE 1 : REVUE DE LA LITTÉRATURE	12
CHAPITRE 2 : FUSION DE SURFACES PAR LA MÉTHODE DE REPRÉSENTATION IMPLICITE	19
2.1 Fonction de champ.....	19
2.1.1 Définition de la fonction de champ	19
2.1.1.1 Calcul automatique de la résolution de la grille	28
2.1.2 Recherche du point le plus près sur la surface.....	32
2.1.3 Optimisations du temps de calcul.....	40
2.1.3.1 Éléments utiles	40
2.1.3.2 Plan de projection des triangles.....	41
2.1.4 Implémentation informatique.....	47
2.1.5 Opérations dans le domaine implicite	51
2.2 Intégration des fonctions de champ.....	53
2.2.1 Principe de l'intégration.....	53
2.2.2 Étapes de l'intégration	55
2.2.2.1 Détection des éléments utiles	55
2.2.2.2 Détection des minimums.....	56
2.2.2.3 Sélection des éléments pour le calcul.....	57
2.2.2.4 Calcul de la valeur intégrée.....	58
2.2.3 Implémentation informatique.....	60
CHAPITRE 3 : TRIANGULATION DU MODÈLE FINAL	64
3.1 Triangulation « marching triangle »	64
3.1.1 Algorithme du « marching triangle »	64

3.1.2	Implémentation informatique.....	76
3.1.3	Améliorations à l'algorithme	93
3.1.3.1	Test de la sphère	94
3.1.3.2	Ordre de traitement des arêtes.....	97
3.1.3.3	Nouveau triangle possible.....	100
3.1.3.4	Efficacité de l'algorithme amélioré.....	101
3.2	Triangulation « marching cube ».....	103
3.2.1	Algorithme du « marching cube ».....	103
3.2.2	Implémentation informatique.....	105
3.2.3	Améliorations à l'algorithme	109
3.2.3.1	Triangles incohérents	110
3.2.3.2	Triangles supplémentaires.....	110
CHAPITRE 4 : ANALYSE DES RÉSULTATS		113
4.1	Montage expérimental	113
4.2	Fusion de surfaces planes.....	116
4.3	Fusion sans redondance	118
4.4	Analyse de surfaces typiques	120
4.4.1	Trous dans les surfaces	121
4.4.2	Triangles isolés	122
4.4.3	Courbes de niveaux.....	124
4.5	Résultat global	126
4.6	Erreur induite par l'algorithme.....	129
4.7	Performances de l'algorithme	133
4.7.1	Distributions des périmètres.....	136
CONCLUSION		138
RECOMMANDATIONS		142
ANNEXE 1 : Fiche technique du MapScan		148
ANNEXE 2 : Principaux articles utilisés.....		152
RÉFÉRENCES BIBLIOGRAPHIQUES.....		184
Tableau I	Erreur induite par le processus de fusion.....	131
Tableau II	Informations numériques sur les surfaces	134
Tableau III	Informations numériques sur l'algorithme de fusion	134

LISTE DES FIGURES

	Page
Figure 1.1 Schéma bloc du MapScan	1
Figure 1.2 Prototype du profilomètre laser	2
Figure 1.3 Système de positionnement	3
Figure 1.4 Principe de fonctionnement d'un profilomètre	5
Figure 1.5 Tâches du bloc d'acquisition et de traitement	8
Figure 1.6 Étapes de la fusion des surfaces	9
Figure 1.7 Interface du logiciel MapScan.....	10
Figure 2.1 Surface et sa grille pour la fonction de champ	20
Figure 2.2 Distance minimale entre un élément et la surface	21
Figure 2.3 Contour d'une surface dans l'espace 3D	23
Figure 2.4 Représentation graphique d'une fonction de champ.....	24
Figure 2.5 Convention du coin inférieur des éléments pour le calcul	26
Figure 2.6 Graphique de la résolution en fonction du périmètre moyen	30
Figure 2.7 Projection orthogonale d'un point sur un plan en 3D	33
Figure 2.8 Vecteurs du point projeté aux sommets du triangle.....	35
Figure 2.9 Projection d'un point sur une droite en 3D.....	36
Figure 2.10 Procédure de recherche du point le plus près sur un triangle.....	39
Figure 2.11 Éléments utiles d'une fonction de champ.....	40
Figure 2.12 Différents plans de projection des triangles.....	42
Figure 2.13 Projection d'un triangle sur le plan de projection.....	43
Figure 2.14 Plan de projection des triangles et séquence de recherche.....	45
Figure 2.15 Filtrage d'une surface implicite.....	52
Figure 2.16 Principe d'intégration de deux fonctions de champ.....	54
Figure 2.17 Intégration de deux surfaces.....	62
Figure 3.1 Principe du « marching triangle »	65
Figure 3.2 Détection d'un nouveau sommet possible	67
Figure 3.3 Sphère circonscrite à un triangle	69

Figure 3.4	Nouveaux triangles potentiels à ajouter à la surface.....	72
Figure 3.5	Étapes de l'algorithme du « marching triangle »	74
Figure 3.6	Progression du « marching triangle » sur un plan	75
Figure 3.7	Caractéristique d'un triangle équilatéral.....	77
Figure 3.8	Relation entre la hauteur et les côtés d'un triangle équilatéral	79
Figure 3.9	Projection d'un nouveau point	81
Figure 3.10	Paramètres utiles au calcul du centre de la sphère.....	83
Figure 3.11	Discrétisation du plan d'un triangle	86
Figure 3.12	Configurations des arêtes de contour voisines.....	88
Figure 3.13	Continuité de la surface entre les triangles	90
Figure 3.14	Sommet de contour le plus près de l'arête en traitement	92
Figure 3.15	Cas problématique du test de la sphère	94
Figure 3.16	Nouvelle sphère pour le test.....	96
Figure 3.17	Nouvel objet potentiel pour le test	97
Figure 3.18	Surface modifiée par l'ordre de traitement des arêtes.....	98
Figure 3.19	Vérification de tous les sommets de contour.....	100
Figure 3.20	Triangulation des voxels	104
Figure 3.21	Convention des indices d'un voxel et exemple de triangulation.....	106
Figure 3.22	Règle de la main droite pour l'orientation du vecteur normal	108
Figure 3.23	Éléments voisins de signe opposé	111
Figure 4.1	Photographie du montage expérimental	114
Figure 4.2	Photographie de la statue de Sophocle	115
Figure 4.3	Résultats de fusion sur un plan.....	117
Figure 4.4	Résultat de fusion sur un morceau de polystyrène.....	119
Figure 4.5	Résultats de fusion sur des portions de la statue de Sophocle	120
Figure 4.6	Principe des triangles isolés sur la surface	123
Figure 4.7	Phénomène des courbes de niveaux	125
Figure 4.8	Résultat de fusion du visage complet de Sophocle.....	127
Figure 4.9	Résultat de fusion du visage de Sophocle vue de profil	128
Figure 4.10	Résultat de fusion sur le moulage d'un pied.....	129
Figure 4.11	Distribution des périmètres pour le moulage du pied	137

INTRODUCTION

Ce projet de maîtrise a été réalisé à l'INO, un institut de recherche en optique situé à Sainte-Foy (Québec), au sein du secteur des systèmes optiques et numériques, dans le cadre d'une collaboration entre l'INO et l'École de technologie supérieure. L'INO a développé un appareil portatif permettant de numériser en trois dimensions (3D) la surface d'un objet solide. Ce projet faisait aussi l'objet d'une collaboration entre l'INO et l'Agence spatiale canadienne. Cette dernière voit un intérêt à la vision numérique pour des applications de numérisation 3D d'objets dans l'espace. D'autres applications plus conventionnelles sont entrevues pour le capteur 3D portable, dont la numérisation d'objets pour l'inspection, la production rapide de modèles 3D pour la conception assistée par ordinateur (CAO), la modélisation et la réalité virtuelle.

L'appareil développé porte le nom de MapScan qui signifie « Manual And Portable Scanner ». Le MapScan est constitué d'un profilomètre laser couplé à un système de positionnement, lesquels sont reliés à un ordinateur. Une application informatique permet de faire l'acquisition, le traitement, l'affichage et la conversion des points et surfaces en modèles polygonaux. La figure 1.1 présente le schéma bloc du MapScan.

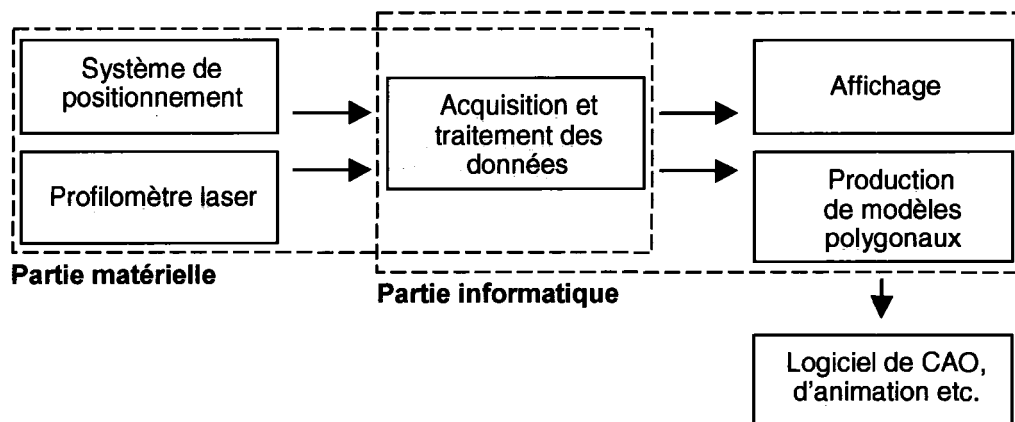
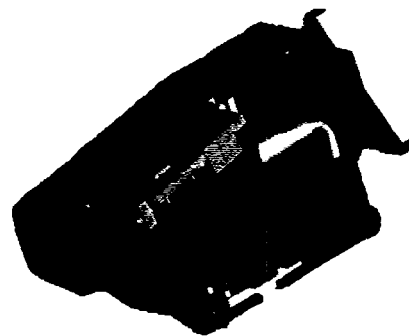


Figure 1.1 Schéma bloc du MapScan

Le profilomètre laser est un capteur optique de profondeur. Il est composé d'une source de lumière structurée (laser couplé à un élément divergent générant une ligne), d'un objectif et d'une caméra CCD. Le faisceau laser divergent est projeté sur un objet. Celui-ci est déformé par l'objet et suit sa surface. La caméra mesure la position du faisceau de lumière dans l'image. Cette dernière est numérisée par une carte d'acquisition spécialisée à l'intérieur de l'ordinateur. Le système de positionnement possède six degrés de liberté et il est basé sur une technologie ultrasonique. Il est composé de deux parties. La première partie comprend trois émetteurs à ultrasons qui sont disposés en forme de triangle sur un plan fixe. La seconde partie comprend trois microphones qui sont aussi disposés en forme de triangle sur un plan mobile. La vitesse de propagation étant connue, en calculant les temps de vol des ondes sonores pour chaque combinaison d'émetteur et de récepteur, il est possible de déterminer la position (X, Y, Z) et l'orientation (tangage, roulis, lacet) du triangle mobile avec les récepteurs par rapport au triangle fixe avec les émetteurs. Le triangle mobile est fixé à l'arrière sur le boîtier du profilomètre. Celui-ci est tenu à la main et il ressemble à un caméscope. La figure 1.2 présente l'aspect du profilomètre avec le récepteur du système de positionnement.



A) Photographie



B) Coupe du modèle CAO

Figure 1.2 Prototype du profilomètre laser

Sur la photographie du profilomètre (figure 1.2A), il est possible d'évaluer la dimension de celui-ci par rapport à la main d'une personne. Sur la coupe du modèle CAO (figure 1.2B), la disposition de la caméra à angle et superposée par rapport au laser est illustrée. Dans les deux cas, la position du triangle mobile avec les récepteurs ultrasons est visible à l'arrière. Il est donc possible de balayer la surface d'un objet de façon aléatoire avec le profilomètre à l'intérieur d'un certain volume défini par le système de positionnement. La figure 1.3 présente le système de positionnement fabriqué par Fakespace Inc. et un montage typique montrant son utilisation avec le profilomètre.

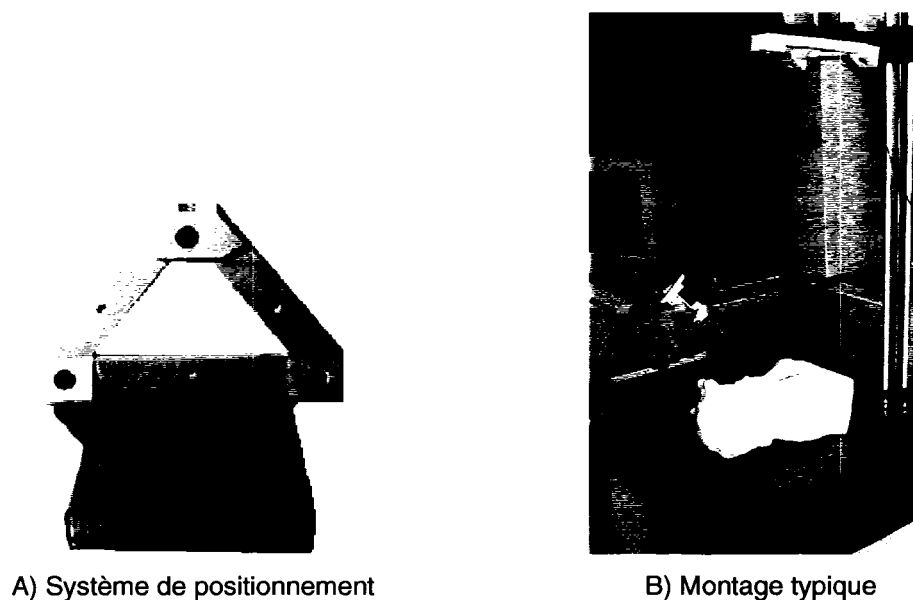


Figure 1.3 Système de positionnement

La figure 1.3A présente les différentes parties du système de positionnement. Il est composé du grand triangle avec les émetteurs qui doit être fixe, du petit triangle avec les récepteurs qui est mobile et qui s'installe sur le profilomètre ainsi que du module électronique. Celui-ci commande les émetteurs et les récepteurs, calcule les distances et les angles des six degrés de liberté et transmet ces informations à l'ordinateur par le port série. À la figure 1.3B le grand triangle émetteur est fixé horizontalement au-dessus de

l'objet à numériser. Le petit triangle récepteur est fixé sur le profilomètre qui est tenu à la main et qui effectue un balayage de la ligne laser sur l'objet.

La partie informatique du MapScan est un programme complet et intégré, développé avec Microsoft Visual C++ utilisant la bibliothèque de classes MFC (Microsoft Foundation Class library). Cette partie est divisée en trois blocs fonctionnels selon la figure 1.1. Le premier est le module logiciel d'acquisition et de traitement de données. Il réalise plusieurs tâches. La première tâche est l'acquisition des données provenant du profilomètre et du système de positionnement. Une image à partir du capteur CCD du profilomètre ainsi que les paramètres des six degrés de liberté du système de positionnement sont acquis à une certaine fréquence (environ 25 Hz dans le cas du présent système). La seconde tâche consiste à traiter l'image pour détecter le pixel dans chaque colonne appartenant à l'image de la ligne laser. L'ensemble des valeurs de position associées aux pixels de cette ligne est appelé un profil. La troisième tâche consiste à associer une profondeur ou une distance à chaque pixel du profil en fonction de sa position dans l'image. Suite à cette opération les pixels du profil contiennent de l'information sur leur profondeur par rapport au référentiel du profilomètre. Jusqu'à présent le traitement sur un profil et les résultats obtenus sont similaires à ceux qui seraient obtenus d'un profilomètre conventionnel où le capteur se déplace linéairement par rapport à la scène à l'aide d'un dispositif comme celui d'un étage de translation.

Comme ce profilomètre est tenu à la main et que sa trajectoire est aléatoire, une quatrième tâche consiste à positionner le profil dans l'espace tridimensionnelle par rapport à un référentiel global de la scène en tenant compte des informations du système de positionnement. Le système de positionnement donne la position et l'orientation du référentiel lié au profilomètre dans le référentiel global pour chacun des profils acquis. Une matrice de transformation est générée pour chacun des profils à partir des informations du système de positionnement et chacun des profils est multiplié par sa matrice pour le positionner dans l'espace du référentiel global. L'information de

profondeur de chaque pixel par rapport au référentiel du profilomètre est donc combinée à la position de celui-ci pour ainsi obtenir la position 3D absolue de chacun des pixels qui ont subi un changement de référentiel et qui sont dorénavant appelés des points 3D ou encore des vertex. Suite à cette étape, les points 3D possèdent des coordonnées tridimensionnelles complètes et ils représentent des points réels sur la surface de l'objet numérisé. Ces données ne sont plus des données de profondeur comme c'était le cas avant l'exécution de la quatrième tâche, elles sont référencées de façon uniforme grâce à un système de référence global. La figure 1.4 illustre le principe de fonctionnement d'un profilomètre.

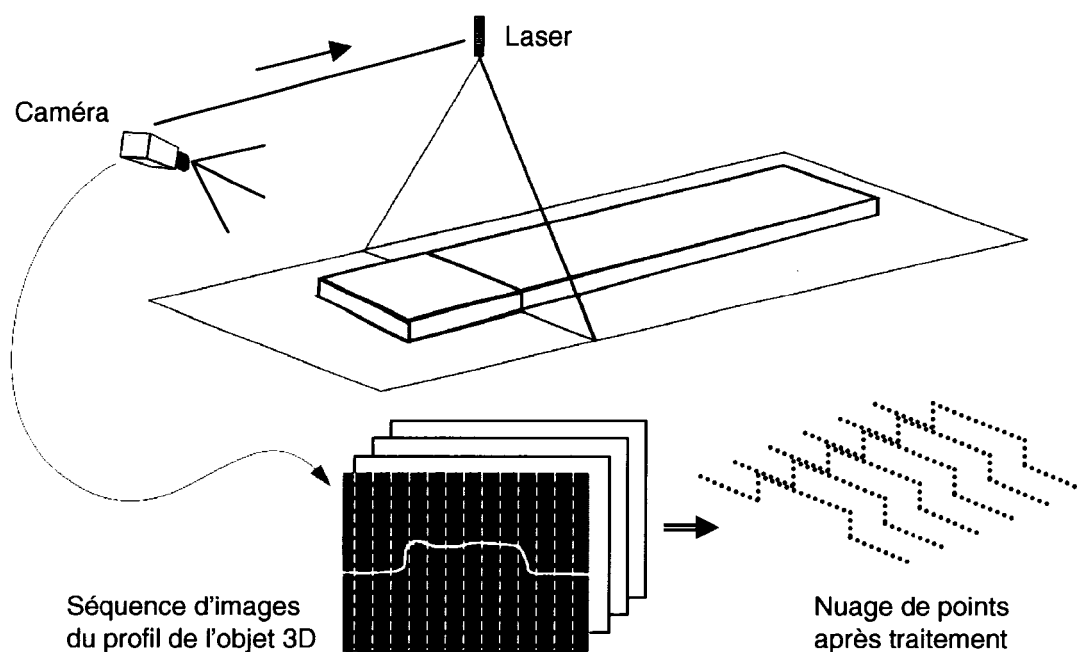


Figure 1.4 Principe de fonctionnement d'un profilomètre

À la figure 1.4, un faisceau laser divergent est projeté sur la surface d'un objet 3D. La caméra regarde la scène sous un angle différent de celui du laser. Une séquence d'images est acquise en déplaçant l'ensemble composé de la caméra et du laser par rapport à l'objet qui est fixe. Une image numérique est composée de pixels disposés en

rangées et en colonnes sur une grille régulière. Les colonnes de l'image sont schématisées à la figure 1.4 par les traits pointillés sur l'image du profil de l'objet. Les paramètres de la caméra sont ajustés pour mettre en évidence la ligne laser sur les images. Pour chaque colonne de l'image, le pixel ayant l'éclairement lumineux maximal du profil laser est détecté. En fonction de la rangée où se trouve ce pixel, une table de correspondance établie par l'étalonnage du profilomètre permet d'associer une profondeur à ce pixel. À la figure 1.4, cette profondeur représente l'épaisseur de l'objet 3D. Lorsque le traitement d'une image est complété, un profil discret composé de points est obtenu. À la suite du traitement sur une séquence d'images, plusieurs profils sont obtenus et un nuage de points représentant la surface de l'objet est ainsi formé.

Le profilomètre utilisé dans le cadre de ce projet est employé pour balayer la surface d'un objet avec la ligne laser et fournir l'information sur la distance. Plusieurs balayages sont nécessaires afin de numériser la surface complète d'un objet. Lorsqu'un balayage est complété, il contient un ensemble de profils dont chacun contient un ensemble de points 3D. Le résultat final peut être considéré comme un nuage de points 3D non structurés, représenté par l'ensemble des vertex. Chaque vertex a une position 3D arbitraire et il n'est pas à distance constante par rapport à ses voisins comme c'est le cas par exemple pour une image de profondeur captée par une caméra 3D à balayage laser synchrone. Les pixels qui appartiennent au profil laser d'une image acquise sont à distance constante sur un seul axe de l'image. Cette caractéristique est perdue lors du changement de référentiel. Deux pixels voisins dans un profil ne sont plus nécessairement des vertex voisins immédiats en terme de distance minimale dans le nuage de points 3D d'un balayage. Le nuage de points possède tout de même une certaine structure hiérarchique étant donné que chaque vertex est référencé à une position connue à l'intérieur d'un profil distinct qui lui aussi est référencé à une position connue à l'intérieur d'un balayage distinct. Donc un vertex peut être perçu comme le $X^{\text{ième}}$ vertex du $Y^{\text{ième}}$ profil du $Z^{\text{ième}}$ balayage de l'objet en cours de numérisation.

La cinquième tâche du bloc acquisition et traitement consiste à réaliser un maillage polygonal de type triangulaire à partir du nuage de points d'un même balayage. Suite à l'opération de maillage ou de triangulation, le nuage de points original se compose maintenant d'une surface dans l'espace 3D formée de plusieurs polygones et plus précisément de triangles. Le maillage contient des triangles composés d'arêtes et de vertex qui sont reliés entre eux pour former les triangles et la surface. Le maillage est réalisé en commençant par filtrer les vertex et les profils pour éliminer la redondance et ensuite en reliant les vertex entre eux pour former les triangles. Cette tâche tient compte de la connectivité entre les vertex et les profils pour obtenir un maillage de bonne qualité.

Dans le cas de la caméra 3D portable, le nuage de points qui est transformé en maillage représente une partie de la surface totale de l'objet à numériser à cause de la restriction du champ de vue. Plusieurs balayages sont effectués pour couvrir l'ensemble de la surface d'un objet. Typiquement chaque balayage qui résulte en une surface partielle de l'objet doit se superposer légèrement à ses limites avec ses voisins de façon à couvrir la surface de l'objet en entier sans laisser de sections non numérisées entre les surfaces partielles. Cette procédure engendre donc une certaine redondance dans la représentation de la surface de l'objet aux endroits où au moins deux surfaces partielles se superposent.

La sixième et dernière tâche du traitement consiste à fusionner toutes les surfaces partielles distinctes de l'objet entre elles pour obtenir une seule surface complète sans redondance représentant l'objet numérisé. Cette dernière tâche non triviale et relativement complexe fait l'objet de ce mémoire. Elle est divisée en trois étapes distinctes qui seront détaillées dans les chapitres suivants. La première étape consiste à calculer une fonction de champ individuelle pour chaque surface partielle. Une fonction de champ est une représentation implicite d'une surface dans l'espace 3D. La seconde étape consiste à fusionner les fonctions de champ individuelles en une seule fonction de

champ global. La troisième étape consiste à produire un maillage de la fonction de champ globale de façon à obtenir une surface explicite composée de triangles dans l'espace 3D. La figure 1.5 résume les différentes tâches du premier bloc fonctionnel d'acquisition et de traitement des données de la partie informatique sous forme d'un ordinogramme.

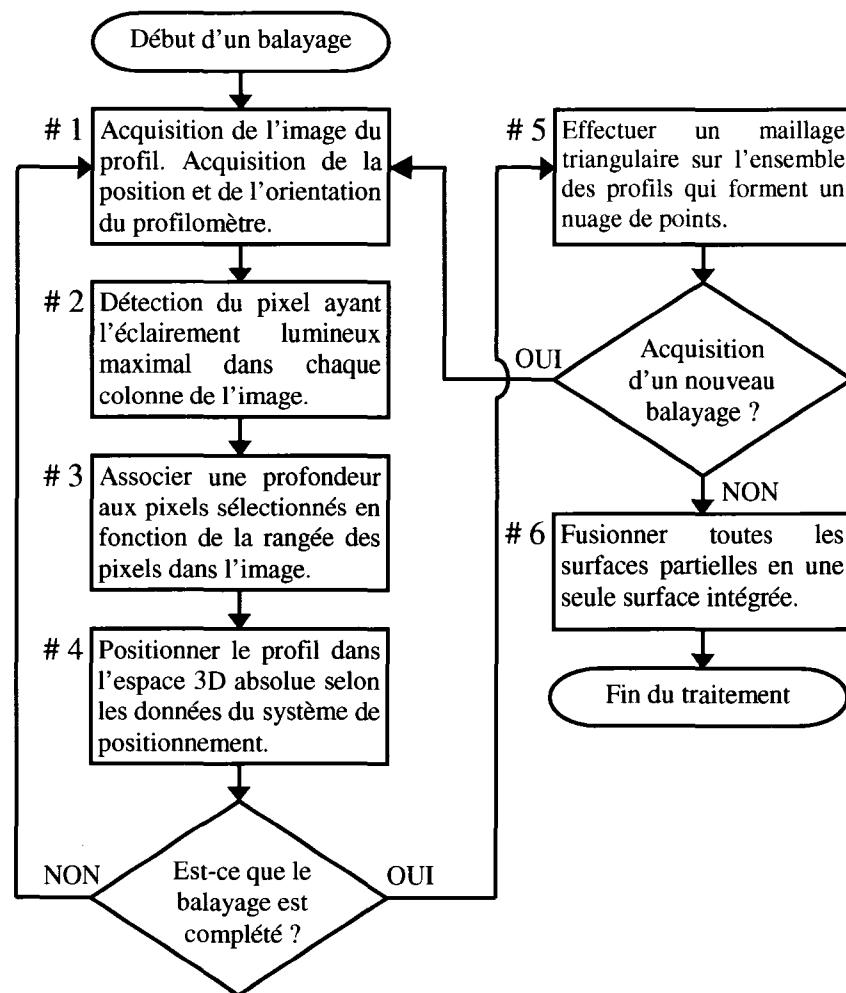


Figure 1.5 Tâches du bloc d'acquisition et de traitement

La figure 1.6 présente, sous forme d'un ordinogramme, les étapes de la sixième tâche de fusion des surfaces indiquée à la figure 1.5 et qui fait l'objet de ce mémoire.

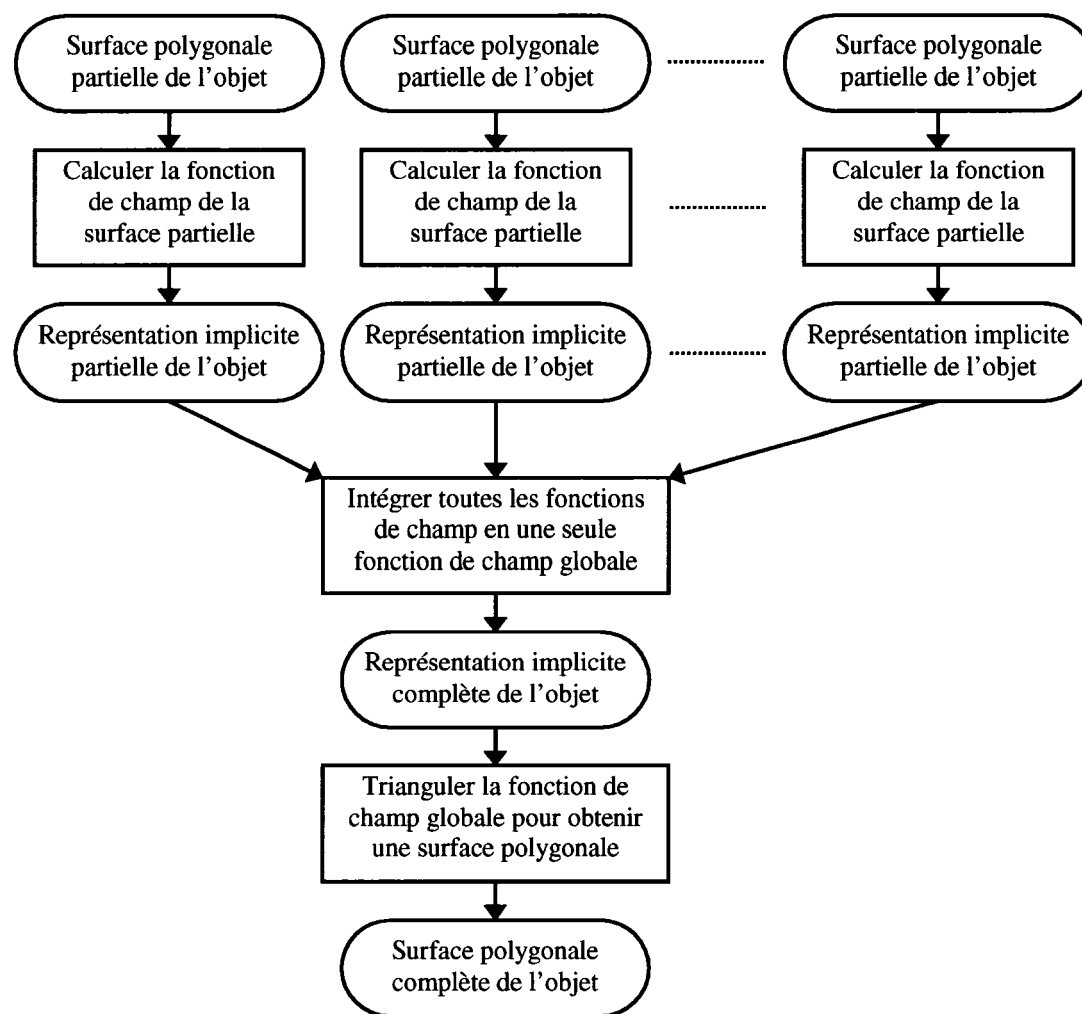


Figure 1.6 Étapes de la fusion des surfaces

Le second bloc fonctionnel de la partie informatique selon la figure 1.1 est celui de l'affichage qui utilise une fenêtre graphique opérant avec des fonctions OpenGL (standard établi par Silicon Graphics Inc) pour afficher les surfaces et les rendus sous différents formats. Le nuage de points d'un balayage est affiché en temps réel lors de l'acquisition. Une fois le balayage terminé, le filtrage et le maillage sont effectués sur le nuage de points et la surface résultante est affichée. Ce processus est répété pour chaque balayage nécessaire à la numérisation complète de la surface de l'objet. Ensuite la

fusion des surfaces est calculée et la surface finale résultante est affichée. La figure 1.7 présente l'interface du logiciel du MapScan.

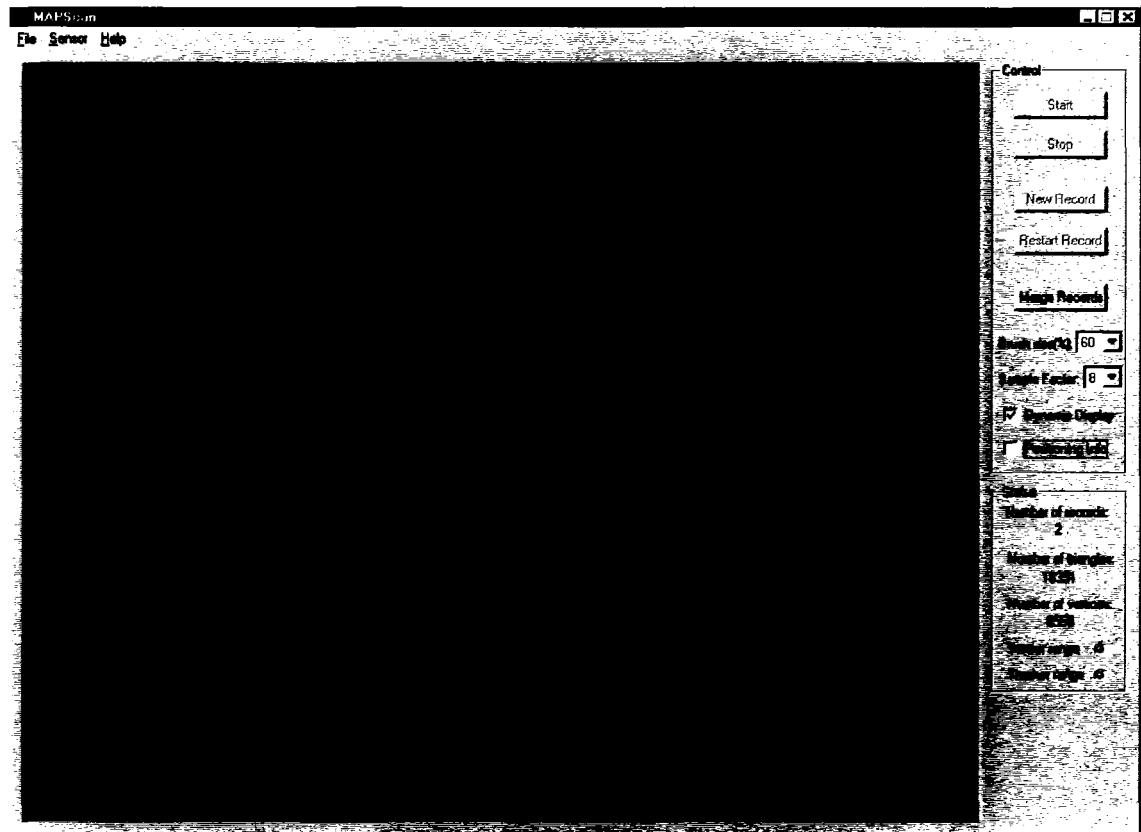


Figure 1.7 Interface du logiciel MapScan

L'objet numérisé dans l'exemple à la figure 1.7 est une statue de la tête de Sophocle, un poète tragique grec (495-406 av. J.-C.). Cet exemple compte seulement deux balayages qui ont été effectués horizontalement. Seul le visage de la statue est représenté. Chaque balayage est affiché d'une couleur différente pour les distinguer facilement. À partir du moment où au moins deux surfaces ont été numérisées, il est possible d'appuyer sur le bouton « Merge Records » qui lance le processus de fusion des surfaces.

Le troisième et dernier bloc fonctionnel est celui de la conversion des surfaces. La surface finale, représentant l'ensemble de l'objet numérisé, est générée par le MapScan dans un format de données particulier. Il s'agit d'un fichier portant l'extension « i3d ». Cette structure de données a été développée à l'INO pour les besoins du MapScan dans le but d'emmagasiner les points 3D et les surfaces de façon efficace. Ce format de données n'est pas compatible avec aucun autre logiciel. Il était donc souhaitable de convertir le format interne de surface en deux formats compatibles avec les logiciels de CAO. Il s'agit du format ASCII et du format STL (stéréolithographique). Le format ASCII est simplement une suite de coordonnées des vertex de la surface dans un fichier texte. Le format STL est un format qui décrit les surfaces d'un objet au moyen de triangles. Il s'agit d'un fichier portant l'extension « stl » qui est compatible avec la plupart des logiciels de CAO, d'animation etc. Il est donc possible d'importer les surfaces numérisées avec des logiciels commerciaux pour différentes applications.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

Une recherche exhaustive a été réalisée sur la base de données spécialisée INSPEC qui contient la plupart des périodiques scientifiques et des comptes rendus de conférences. Suite à quelques essais pour trouver une formule de recherche efficace, la chaîne de mots clés qui a été utilisée pour faire cette recherche est la suivante : « 3D and (image or surface) and (fusion or merging) ». Cette recherche a permis de trouver des centaines d'articles reliés à la reconstruction 3D de surfaces. Une première lecture de ces articles a permis d'éliminer tous ceux qui n'avaient pas de lien avec la problématique de ce mémoire. Suite à une deuxième lecture plus approfondie des articles retenus, plusieurs méthodes potentielles pour intégrer plusieurs surfaces en une seule ont été ciblées. Elles ont été classées en différentes familles selon leur principe de fonctionnement.

Cependant la plupart de ces méthodes ont été testées sur des images de profondeur qui sont considérées comme étant des données 2,5D et non sur des vraies données 3D comme celles provenant du MapScan. Une image de profondeur est semblable à une image à deux dimensions (2D) où chaque pixel ou élément est sur une grille ordonnée dans un plan. La seule différence est que pour une image 2D, l'information qui est codée à chaque élément de l'image (pixel) est la valeur d'éclairement lumineux du pixel alors que pour une image de profondeur 2,5D, cette information est la distance dans la troisième dimension entre un point de vue unique et le point d'intérêt sur l'objet. Il y a une grande différence entre ce type de données et le nuage de points 3D d'un balayage du MapScan. Les points 3D d'un balayage ne sont pas ordonnés sur aucune grille et ils ne sont pas non plus référencés par rapport à un point de vue. Ils représentent vraiment des coordonnées 3D complètes et indépendantes sur la surface de l'objet d'intérêt.

Une première famille de méthodes de reconstruction 3D de surfaces regroupe les méthodes volumétriques [(Algorri & Schmitt, 1996), (Bajaj & al, 1995), (Curless & Levoy, 1996), (Hilton & al, 1996b), (Hoppe & al, 1992), (Roth & Wibowo, 1995), (Wheeler & al, 1996)] qui commencent par diviser l'espace 3D en des éléments de base appelés des voxels. Habituellement le référentiel utilisé est en coordonnées rectangulaires et un voxel est représenté par un petit cube élémentaire. Plus le cube est petit, plus la résolution augmente ainsi que la quantité de mémoire nécessaire et le temps de calcul. Il faut donc trouver un juste milieu en fonction des besoins de l'application. Chaque élément d'une image de profondeur est placé dans un voxel respectif selon sa position. Cette procédure est appliquée à toutes les images de profondeur qui servent à décrire l'objet en entier. Ensuite, pour tous les points à l'intérieur d'un même voxel, la redondance est éliminée en utilisant une technique simple. Par exemple, tous les points peuvent être éliminés et le voxel est marqué afin de l'identifier comme faisant partie de l'objet. Ou encore, pour obtenir une résolution sous-voxel, une moyenne peut être calculée pour déterminer l'emplacement du point résultant de la surface en fonction de tous les points considérés à l'intérieur du voxel. Les points résultants sont reliés entre eux avec un algorithme de triangulation comme celui du « marching cube » (Lorensen & Cline, 1987).

Ces méthodes fonctionnent aussi bien avec des images de profondeurs dont seulement les points sont disponibles comme décrit précédemment ou encore avec des images de profondeurs qui ont été préalablement maillées individuellement. Dans ce dernier cas, ce sont les sommets des triangles qui sont placés dans les voxels et l'information sur la connectivité des points et leur vecteur normal à la surface est conservée pour améliorer l'efficacité de l'élimination de la redondance. Ces méthodes ont toutes été testées sur des images de profondeurs. Cependant l'idée de diviser l'espace 3D en éléments de base (voxel) et de remplir ces éléments avec les points des surfaces en conservant l'information sur la géométrie locale des surfaces semble intéressante et compatible avec le type de surfaces produit par le MapScan.

Une seconde famille regroupe les méthodes surfaciques [(Pito, 1996), (Rutishauser & al, 1994), (Soucy & Laurendeau, 1995a), (Soucy & Laurendeau, 1995b), (Turk & Levoy, 1994)] qui débutent en maillant les images de profondeurs individuellement. Ensuite toutes les surfaces sont alignées en utilisant un algorithme comme celui du « iterative closest point » (ICP) (Besl & McKay, 1992). Les zones de redondance dans les surfaces sont étiquetées en utilisant de l'information provenant de l'algorithme ICP ou encore avec un critère de distance fixe. Ces zones sont éliminées par une forme d'érosion des surfaces ou par une autre technique similaire. Les contours des surfaces érodées résultantes sont détectés et reliés entre eux en ajoutant des triangles aux jonctions où les contours sont assez près les uns des autres pour former la surface résultante.

Ces méthodes surfaciques ne modifient donc pas les portions de surfaces dans les zones non redondantes. En d'autres termes, la surface résultant de l'intégration des surfaces partielles de l'objet aura les mêmes triangles aux mêmes endroits que ceux des surfaces partielles dans les zones non redondantes entre elles. L'algorithme agit seulement dans les zones redondantes en les éliminant et en tissant des liens entre les zones non redondantes restantes. À l'inverse, les méthodes volumétriques maillent la surface résultante au complet et les triangles originaux des surfaces partielles ne sont pas conservés dans le résultat final. Il est difficile d'évaluer si cette caractéristique est souhaitable ou non à ce stade-ci. À première vue, si deux surfaces ayant une zone de redondance ont un espace de désalignement, les méthodes volumétriques auront probablement tendance à moyenniser cet espace sur toute la zone de redondance alors que les méthodes surfaciques auront probablement tendance à créer un échelon à la jonction entre les deux surfaces.

Ces méthodes surfaciques semblent moins intéressantes que les méthodes volumétriques pour différentes raisons. Le critère de distance à utiliser pour déterminer s'il y a redondance ou non doit tenir compte de la précision du capteur utilisé. Si cette

précision est faible, l'élimination de la zone de redondance laissera un espace assez grand entre les surfaces. Il sera sûrement difficile de joindre les surfaces en générant de nouveaux triangles dans ce cas et ces triangles générés aux jonctions seront sûrement plus gros que la moyenne des autres triangles de la surface. De plus, les procédures pour éliminer les zones de redondance ne semblent pas tenir compte de certains aspects afin de conserver des surfaces consistantes sans introduire ou agrandir des trous ou d'autres imperfections de surface.

Bien que ces méthodes aussi aient toutes été testées sur des images de profondeur, il semble qu'elles soient aussi compatibles avec les surfaces du MapScan. Cependant l'étape de l'alignement des surfaces ne serait pas utile dans le cas des surfaces du MapScan. Le désalignement des images de profondeur vient du changement de point de vue d'une image à une autre. Il est donc nécessaire d'aligner les différentes surfaces en se basant sur les détails communs entre les images. Les surfaces du MapScan sont par défaut déjà alignées car elles sont référencées à l'aide du système de positionnement en temps réel lors du balayage. Contrairement aux autres étapes de ces méthodes surfaciques, les algorithmes pour créer les nouveaux triangles aux jonctions sont toutefois vraiment adaptés pour fonctionner avec des images de profondeur.

Une troisième méthode basée sur des modèles solides (Reed & Allen, 1997) a été répertoriée. Cette méthode débute par le maillage des images de profondeur. Ensuite chaque surface est déplacée d'une certaine distance dans la direction du point de vue de son image de profondeur. Un modèle solide 3D est construit à partir de la surface originale, de sa translation et du volume compris entre les deux. Donc il y a autant de solides qu'il y a d'images de profondeur. L'intersection entre tous ces solides forme un autre solide et le contour de celui-ci représente une surface fermée qui décrit l'objet d'intérêt. Cette surface est donc maillée pour obtenir la surface polygonale finale. Cette méthode ne s'applique pas vraiment aux surfaces du MapScan car son idée principale

est la translation pour produire un solide et il n'y a pas d'équivalent au point de vue d'une image de profondeur dans les surfaces du MapScan.

Une quatrième méthode par surface paramétrique (Motavalli & al, 1998) commence par effectuer une approximation de type B-spline bi-cubique sur chaque image de profondeur pour paramétrer les surfaces individuellement. Ensuite ces surfaces paramétriques sont extrapolées pour s'assurer d'obtenir des intersections entre elles. Ces intersections sont calculées mathématiquement et une représentation paramétrique intégrée pour toutes les surfaces est déterminée. Cette dernière représentation est le résultat final de cette méthode qui ne s'applique pas vraiment aux données du MapScan. Les points de la grille régulière d'une image de profondeur sont utilisés comme points de contrôle pour générer la surface paramétrique. Il serait sûrement possible de générer une surface paramétrique à partir d'un balayage du MapScan mais le traitement serait beaucoup plus complexe que celui à partir d'une image de profondeur. Cette méthode nécessite aussi une intervention humaine pour spécifier les bornes des régions où calculer numériquement les intersections entre les surfaces paramétriques afin que l'algorithme converge adéquatement. Ceci n'est pas souhaitable pour le logiciel du MapScan qui doit faire tout le traitement de façon automatique. De plus il faudrait implanter un post-traitement pour mailler de façon explicite la surface paramétrique résultante.

Une cinquième méthode par surface implicite [(Hilton & al, 1998), (Hilton & al, 1996c)] débute par le maillage individuel des images de profondeur. Ensuite les surfaces explicites sont transformées dans un autre domaine de représentation pour obtenir des surfaces implicites. Ce domaine de représentation est celui des fonctions de champ scalaires 3D. Il s'agit de définir une grille 3D autour de la surface. À chaque élément de la grille la distance minimale à la surface est calculée. Toutes les fonctions de champ individuelles sont intégrées en une seule fonction de champ globale par un processus de fusion. Celle-ci est triangulée avec un algorithme de triangulation comme

celui du « marching triangle » [(Hilton & Illingworth, 1997), (Hilton & al, 1996a)] pour obtenir la surface résultante de façon explicite. Cette méthode a aussi été testée avec des nuages de points 3D similaires à ceux générés par le MapScan et elle semble donner de bons résultats. Elle s'apparente un peu aux méthodes volumétriques étant donné que dans les deux cas, l'espace 3D est divisé en éléments de base ou en volume élémentaire et qu'à chacun de ces éléments de l'information sur les surfaces est calculée. C'est aussi à partir de l'information contenue dans ces éléments que la redondance entre les surfaces est éliminée. La triangulation finale est aussi faite à partir de la représentation volumétrique dans les deux cas. Cette méthode semble très intéressante et compatible avec les données du MapScan.

Plusieurs méthodes ont donc été développées par des chercheurs pour fusionner des surfaces partielles en une seule surface globale. La plupart de ces méthodes utilisent des images de profondeur comme type de données à traiter. Seulement quelques méthodes pourraient être appliquées aux données du MapScan. Celui-ci pourrait être utilisé avec un étage de translation en prenant des points de vue uniques pour rendre les données compatibles avec toutes les méthodes. Cette pratique n'est cependant pas souhaitable puisque le but d'un capteur tenu à la main est justement d'avoir une grande liberté de mouvements lors des balayages.

Considérant la présente approche d'un capteur 3D portable tenu à la main, la méthode à utiliser pour effectuer la fusion des surfaces doit rencontrer plusieurs spécifications. La méthode doit être en mesure de traiter les données 3D provenant du MapScan, c'est à dire des points 3D et des surfaces non structurées avec de la redondance entre elles. Elle doit aussi être capable de composer avec un alignement non parfait des surfaces qui provient de l'erreur du système de positionnement. La méthode doit être la plus robuste possible pour être immunisée aux différents bruits comme le bruit optique du capteur ou encore comme celui des trous dans les surfaces aux endroits où il manque des données.

Idéalement la précision de la surface résultante devrait être la même que celle des surfaces acquises par rapport à l'objet réel qui a été numérisé.

De toutes les méthodes répertoriées, celle qui est la plus appropriée au traitement à réaliser est la méthode de Hilton par surface implicite [(Hilton & al, 1998), (Hilton & al, 1996c)]. Cette méthode, bien que décrite de façon générale et résumée, est complète et est réalisable concrètement de façon détaillée avec les outils logiciels disponibles pour le projet. De plus le domaine de représentation des fonctions de champ laisse entrevoir la possibilité d'effectuer des opérations de traitement des données 3D autres que celle de la fusion sur les surfaces telles que le filtrage 3D ou des transformations géométriques comme le dimensionnement par facteur d'échelle par exemple. Cette méthode a donc été choisie pour réaliser la fusion des surfaces provenant du MapScan. Son fonctionnement, son implémentation et les modifications qui y ont été apportées sont détaillés dans les chapitres suivants.

CHAPITRE 2

FUSION DE SURFACES PAR LA MÉTHODE DE REPRÉSENTATION IMPLICITE

Ce chapitre présente la méthode de fusion de surfaces par la représentation implicite et il est divisé en deux sections. La première section introduit le domaine de la représentation implicite par la fonction de champ et la seconde traite de l'intégration des fonctions de champ en une seule pour réaliser la fusion des surfaces dans ce domaine.

2.1 Fonction de champ

Cette section présente le concept de la fonction de champ. La fonction de champ est définie et la recherche surfacique nécessaire au calcul de celle-ci est détaillée. Des optimisations élaborées pour diminuer les temps de calcul ainsi que l'implémentation informatique des algorithmes sont décrites dans cette section. Des opérations possibles, autre que l'intégration, sur les surfaces dans ce domaine sont aussi présentées.

2.1.1 Définition de la fonction de champ

La fonction de champ est un outil pour représenter dans un autre domaine une surface polygonale dans l'espace 3D. Dans le cas présent, la surface est composée de triangles. La surface polygonale est dite explicite parce qu'elle est explicitement définie par les triangles qui la composent alors que la fonction de champ est considérée comme une représentation implicite de la surface. Pour calculer la fonction de champ d'une surface en particulier, il faut commencer par définir une enveloppe qui englobe cette surface. Cette enveloppe est orientée selon les axes du système de coordonnées. Dans ce cas-ci, le système de coordonnées utilisé est rectangulaire. Donc l'enveloppe prend la forme d'un parallélépipède rectangulaire qui est appelé la boîte de contour. Il s'agit du parallélépipède minimal orienté en X, Y et Z qui peut contenir entièrement la surface

polygonale. Le volume à l'intérieur de la boîte de contour est ensuite divisé en éléments de base à l'aide d'une grille 3D. Les éléments de base sont des cubes. Comme la boîte de contour n'est pas nécessairement un cube et qu'elle dépend de la surface, il se peut que le nombre d'éléments soit différent dans chacun des axes de la boîte. La grosseur de la grille ou des éléments influence directement plusieurs paramètres du processus. Ce point sera traité à la section suivante. La figure 2.1 présente la grille 3D de la fonction de champ pour une surface donnée.

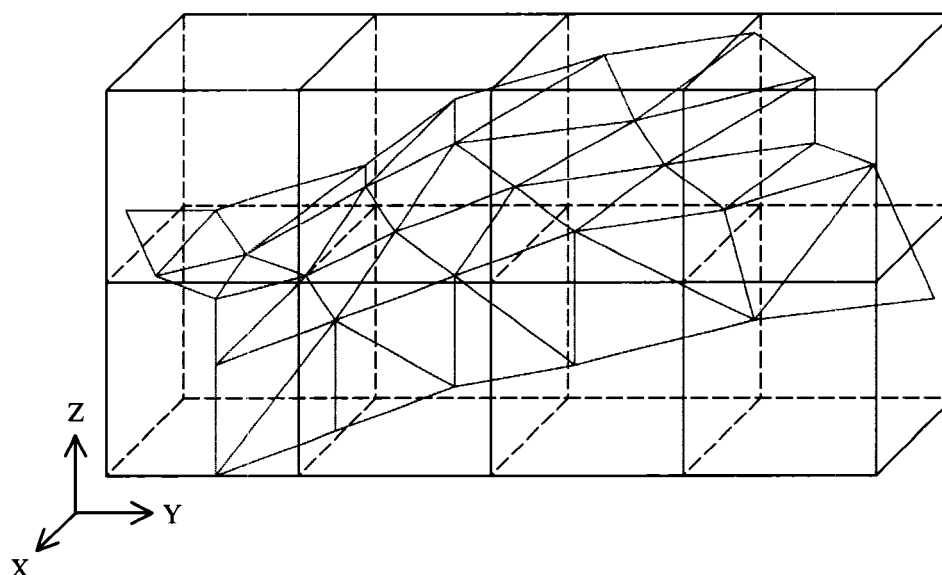


Figure 2.1 Surface et sa grille pour la fonction de champ

À la figure 2.1, la surface composée de triangles est affichée en gris. La boîte de contour de la surface qui définit le domaine spatial de la fonction de champ est affichée en rouge. La grille 3D qui crée les divisions à l'intérieur de la boîte de contour est affichée en bleu. La fonction de champ de l'exemple à la figure 2.1 contient huit éléments, soit un élément selon l'axe X, quatre éléments selon l'axe Y et deux éléments selon l'axe Z.

Une fois la boîte divisée en éléments de base, l'information qui constitue la fonction de champ est calculée à chaque élément. Cette information est essentiellement une distance à la surface. À chaque élément de la grille, la distance minimale à la surface polygonale est calculée. Pour ce faire, le point le plus près de l'élément en question qui se trouve sur la surface est déterminé et la distance entre l'élément et ce point est calculée (la méthode servant à déterminer le point le plus près sera détaillée ultérieurement dans ce chapitre). Un signe positif ou négatif est associé à la distance minimale. Si l'élément de la grille est devant la surface par rapport au point le plus près sur la surface en fonction du vecteur normal à la surface en ce point alors le signe est positif. Si au contraire l'élément est derrière la surface alors le signe est négatif. Lorsque le calcul des distances est complété pour chaque élément, la fonction de champ est déterminée. La figure 2.2 présente le concept de la distance entre un élément et le point le plus près de cet élément sur la surface polygonale.

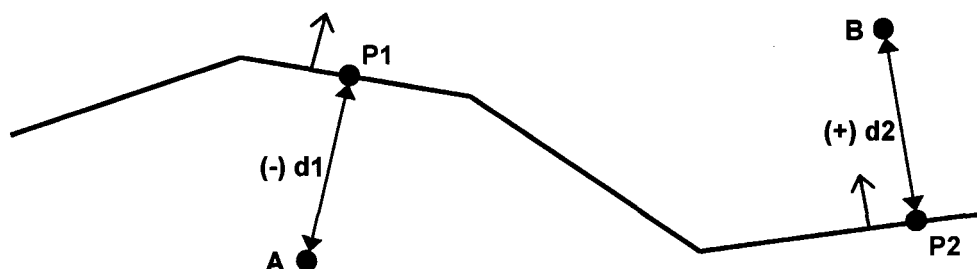


Figure 2.2 Distance minimale entre un élément et la surface

Les segments de droite en noir à la figure 2.2 représente une coupe de la surface. Les points A et B sont des éléments de la fonction de champ et les points $P1$ et $P2$ sont les points les plus près des éléments A et B respectivement sur la surface. Les flèches rouges représentent les vecteurs normaux à la surface aux points $P1$ et $P2$. La distance $d1$ est la distance minimale entre l'élément A et la surface. Cette distance est négative puisque l'élément A est derrière ou en dessous de la surface. La distance $d2$ est la distance minimale entre l'élément B et la surface. Cette distance est positive puisque l'élément B est devant ou au-dessus de la surface.

En se déplaçant d'un élément à l'autre à l'intérieur de la boîte de contour de la surface, il est possible de savoir si l'élément se trouve devant ou derrière la surface et à quelle distance de celle-ci. La surface est donc devenue implicitement définie car la surface polygonale n'existe plus dans ce domaine et seulement le champ scalaire des distances à la surface dans l'espace 3D est disponible pour décrire la surface. À partir de cette information il n'est pas possible de visualiser la surface contrairement à la surface polygonale. Cependant en s'intéressant à tous les éléments où la distance à la surface est nulle, des points de contrôle par où la surface passe exactement sont obtenus. Ceci est uniquement valide en théorie. En pratique, comme les éléments ont une grandeur finie et non infinitésimale et que le domaine est discret, il faudrait plutôt dire que la surface passe par les éléments dont la distance est inférieure à un certain seuil qui devrait être égal à la moitié de la grandeur d'un élément, soit la dimension d'un demi-cube. Pour être plus précis, des points de contrôle sur la surface pourraient être calculés par interpolation entre deux éléments voisins ayant des distances de signe opposé, ceci signifiant que la surface passe entre ces éléments puisque l'un est devant et l'autre est derrière celle-ci. Les points de la surface ainsi obtenus à partir de la fonction de champ sont seulement une approximation ponctuelle de la surface explicite car ceux-ci ne sont pas reliés entre eux pour former une surface continue.

La représentation implicite de la surface offre plusieurs avantages par rapport à la représentation explicite. Entre autres, elle offre la possibilité de faire la fusion de plusieurs surfaces en une seule intégrée. Maintenant que le principe de base de la fonction de champ a été exposé, il faut mentionner que la distance n'est pas la seule information calculée et conservée à chaque élément. Deux informations supplémentaires sont nécessaires au processus de fusion. Il s'agit du vecteur normal à la surface ainsi que d'une information de contour. Comme la distance minimale à la surface est calculée à partir du point le plus près trouvé sur la surface, le vecteur normal à la surface en ce point le plus près est calculé et conservé à chaque élément avec la distance. De plus, le point le plus près trouvé sur la surface peut être sur le contour de la

surface ou non. Cette information de contour est aussi conservée à chaque élément avec la distance et le vecteur normal à la surface. La figure 2.3 illustre le contour d'une surface dans l'espace 3D.

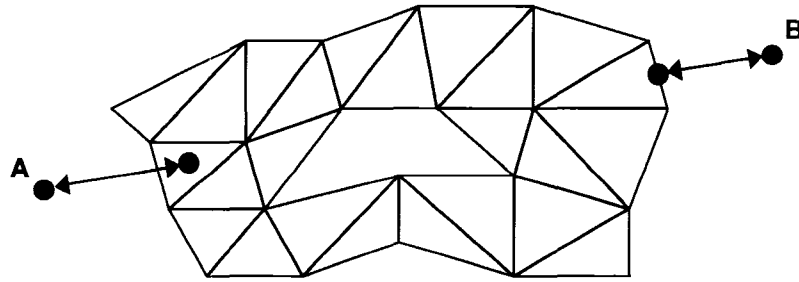
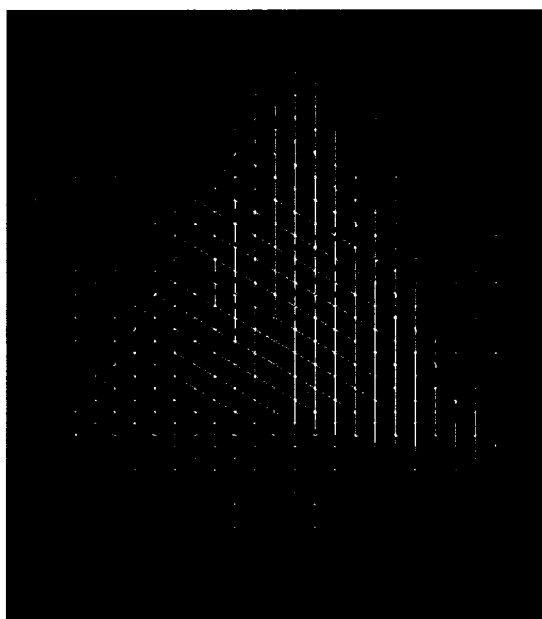
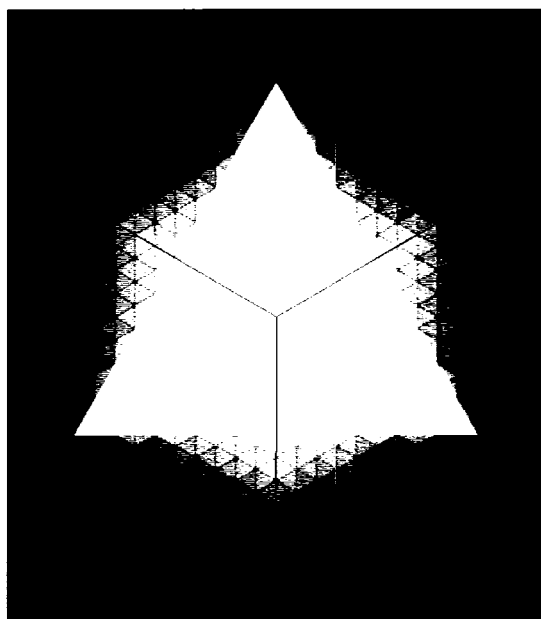


Figure 2.3 Contour d'une surface dans l'espace 3D

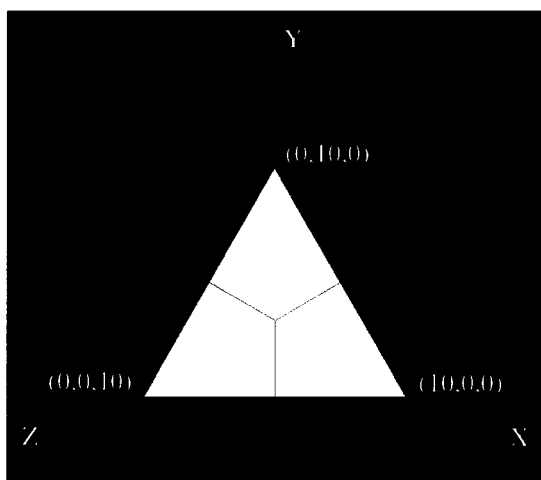
Seulement les arêtes en rouge à la figure 2.3 sont considérées comme le contour de la surface. Si la surface possède une ouverture comme c'est le cas à la figure 2.3, alors le contour de cette ouverture dans la surface est aussi considéré comme étant le contour de la surface. Le point le plus près de l'élément *B* est sur le contour de la surface, cet élément est donc étiqueté comme un élément de contour. Le point le plus près de l'élément *A* n'est pas sur le contour de la surface, cet élément est donc étiqueté comme un élément de non-contour. La figure 2.4 représente une surface ainsi que sa fonction de champ sous différents formats.



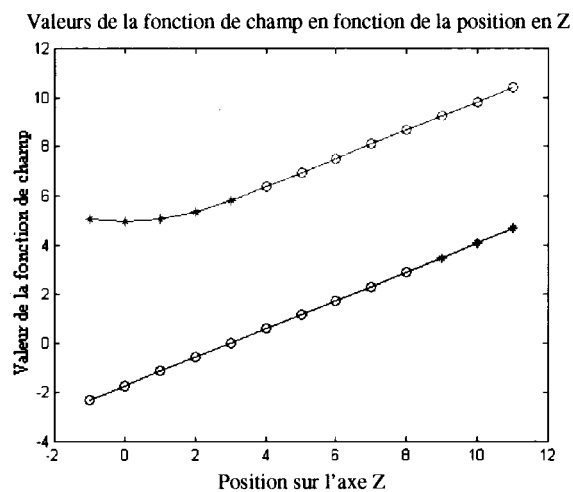
A) Surface et fonction de champ en fil de fer



B) Surface et fonction de champ en solide



C) Surface et système de coordonnées



D) Valeurs de la fonction de champ

Figure 2.4 Représentation graphique d'une fonction de champ

La figure 2.4 contient plusieurs informations intéressantes pour comprendre davantage la fonction de champ. La figure 2.4C présente la surface utilisée dans cet exemple avec son système de coordonnées. Cette surface n'est pas une surface réelle obtenue à l'aide du profilomètre, il s'agit plutôt d'une surface de synthèse qui a été générée

numériquement. Elle est composée d'un seul triangle, représenté en format solide, pour faciliter la compréhension. Les coordonnées des sommets du triangle sont représentées à la figure 2.4C. Toutes les valeurs numériques sont en millimètres. L'axe X est à droite, l'axe Y vers le haut et l'axe Z à gauche selon le point de vue représenté. Ce point de vue, qui est le même que pour les figures 2.4A et 2.4B, est aussi normal à la surface du triangle. Le cube autour du triangle représente la boîte de contour dans laquelle la grille 3D et donc la fonction de champ sont définies.

La figure 2.4A représente la surface et la grille utilisée pour le calcul de la fonction de champ en mode fil de fer où seulement les arêtes des triangles et des cubes sont visibles. Des outils logiciels de visualisation, comme la possibilité d'afficher une fonction de champ, ont été développés spécifiquement pour ce projet afin d'aider à la validation des algorithmes en offrant un support visuel. Ceci a grandement contribué à diminuer le temps de développement du projet. La résolution de la grille dans cet exemple est de un millimètre. Les éléments sont des cubes dont les arêtes ont un millimètre de longueur.

Une convention a été établie pour calculer les valeurs de la fonction de champ. Comme la valeur de la fonction de champ dépend de la localisation dans l'espace d'un élément, la distance à la surface est calculée à partir du coin inférieur dans les trois axes de l'élément et elle est valide pour tout le volume de l'élément. Par exemple, l'élément dont le coin inférieur est à $(1,2,0)$ utilise ces coordonnées pour le calcul de la fonction de champ. Le résultat est valide pour le volume du cube compris entre $(1,2,0)$ et $(2,3,1)$ en excluant ce dernier point maximum puisqu'il sera considéré pour l'élément dont il est le point minimum. Le calcul se fait une seule fois au coin inférieur de l'élément. Pour avoir une surface implicite définie avec plus de précision, il suffit d'augmenter la résolution de la grille en respectant la même convention. La figure 2.5 illustre cette convention.

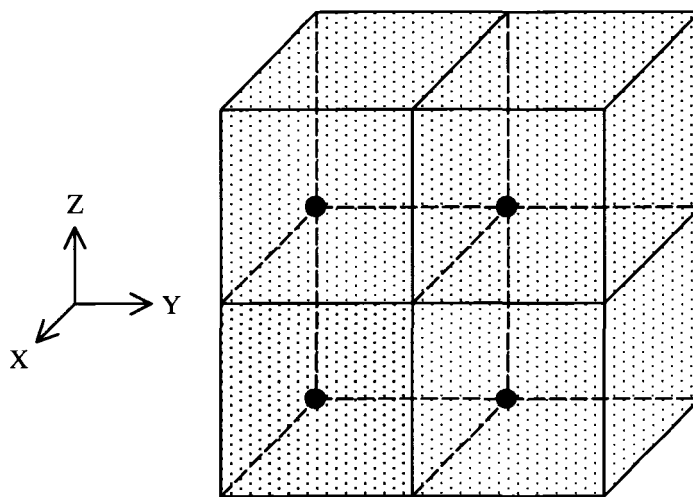


Figure 2.5 Convention du coin inférieur des éléments pour le calcul

À la figure 2.5, la grille de la fonction de champ contient quatre éléments. Les points de calcul utilisés pour évaluer les distances minimales à la surface sont identifiés par les sphères de différentes couleurs. Les valeurs calculées à ces points sont valides pour tout le volume de la même couleur que le point.

Certains éléments de la grille à la figure 2.4A sont affichés en jaune et d'autres en blanc. Les éléments en jaune représentent des valeurs de fonction de champ dont le point le plus près sur la surface qui a servi à calculer la distance minimale est sur le contour de la surface. À l'inverse, les éléments en blanc représentent des valeurs dont le point le plus près n'est pas sur le contour de la surface. Comme le point de vue de l'image est normal à la surface, les éléments en blanc sont tous exactement superposés à la surface alors que ceux en jaune sont plutôt autour de la surface. La figure 2.4B affiche pratiquement la même information que la figure 2.4A mais en mode solide, où toutes les surfaces des triangles et des cubes sont visibles, plutôt qu'en mode fil de fer. Les éléments de la grille sont semi-transparents pour permettre de distinguer la surface.

La figure 2.4D présente un graphique avec deux courbes indépendantes. Pour des valeurs fixes en X et Y, les courbes représentent les valeurs de la fonction de champ en fonction de Z, du minimum au maximum de la boîte de contour. Les positions discrètes des éléments sont identifiées par des cercles ou des croix sur les courbes. Les cercles indiquent que ce ne sont pas des éléments de contour alors que les croix indiquent que ce sont des éléments de contour. La courbe en bleu est pour $(X = 2; Y = 5)$ et la courbe en rouge est pour $(X = 8; Y = 9)$. Les deux rangées d'éléments considérées sur le graphique sont encerclées à la figure 2.4A avec leurs couleurs respectives.

À la figure 2.4D, la plupart des éléments de la courbe bleue ne sont pas des éléments de contour à l'exception des trois derniers étant donné que les projections orthogonales de ceux-ci sur le plan du triangle à la figure 2.4A arrivent à l'extérieur du triangle. La courbe commence avec des valeurs négatives étant donné que les éléments sont en dessous de la surface. Ensuite elle passe par zéro à $Z = 3$, ce qui signifie que la surface passe par ce point $(X = 2; Y = 5; Z = 3)$. Les autres points de la courbe sont composés de valeurs positives qui sont donc au-dessus de la surface. Ces valeurs augmentent avec un déplacement dans le sens croissant de l'axe Z, signifiant un éloignement de la surface. Les cinq premiers éléments de la courbe rouge sont des éléments de contour alors que les projections orthogonales des suivants sur le plan du triangle à la figure 2.4A arrivent à l'intérieur du triangle. Toutes les valeurs de cette courbe sont positives ce qui signifie qu'en se déplaçant sur l'axe Z, les positions obtenues se trouvent toujours au-dessus de la surface.

La représentation d'une surface par la fonction de champ est similaire à celle par des voxels. Les différences sont que les éléments de la grille d'une fonction de champ sont perçus comme des points de l'espace indépendants des autres alors que la représentation par des voxels forme des volumes élémentaires (généralement des cubes) avec des éléments voisins. De plus, les éléments d'une fonction de champ contiennent de l'information sur l'orientation locale ainsi que sur le contour de la surface alors qu'une

représentation par des voxels contient seulement de l'information sur la distance à la surface à partir des sommets des cubes. Selon une représentation conventionnelle par des voxels, il ne serait pas possible de fusionner plusieurs surfaces entre elles essentiellement à cause du manque d'information sur la géométrie locale des surfaces. Cette opération de fusion est toutefois possible avec une représentation par des fonctions de champ.

2.1.1.1 Calcul automatique de la résolution de la grille

Dans l'exemple précédent, pour expliquer la nature de la fonction de champ, une grille de résolution arbitraire d'un millimètre a été utilisée. En travaillant avec des surfaces réelles, il faut se demander quelle résolution doit être utilisée, c'est-à-dire quelle est la dimension appropriée des éléments, soit la longueur des arêtes des cubes de la grille. Ce paramètre influence beaucoup le comportement de l'ensemble des algorithmes décrits dans le présent mémoire. En effet, plus la résolution augmente pour une même surface, plus le nombre d'éléments qui composent la fonction de champ pour un même volume d'une boîte de contour augmente également. Tous les algorithmes traitent un élément à la fois, que ce soit le calcul de la fonction de champ, l'intégration des fonctions de champ ou la triangulation finale. Le nombre plus élevé d'éléments résulte en un accroissement du temps de calcul qui n'est pas linéaire en fonction de la résolution mais plutôt de type cubique puisque le traitement est fait en 3D. Par exemple si la résolution double alors le nombre d'éléments est augmenté de huit fois. Ce facteur n'est pas négligeable et il influence directement les performances de l'ensemble des algorithmes. Étant donné qu'un élément contient plusieurs informations en mémoire, la résolution influence aussi de façon cubique la quantité de mémoire nécessaire aux calculs. La meilleure résolution possible est donc directement reliée aux performances du matériel informatique disponible.

La résolution de la grille influence également la qualité de la surface explicite finale qui a été fusionnée. Sans entrer pour l'instant dans les détails de l'algorithme de

triangulation qui fait l'objet d'un chapitre ultérieur, disons simplement que la grandeur des triangles de la surface résultante est directement liée à la dimension des éléments de départ. Plus la résolution est élevée, plus les triangles seront petits et plus il sera possible d'obtenir des détails précis dans le résultat final. La méthode de caractérisation de la grandeur des triangles utilise l'information provenant des histogrammes du périmètre et de l'aire des triangles d'une surface. Tout le processus de fusion des surfaces utilise en entrée plusieurs surfaces explicites polygonales composées de triangles. Les triangles de ces surfaces ont une grandeur moyenne qui a été déterminée lors des étapes précédentes à la fusion. Ce processus de détermination de la dimension des triangles s'effectue entre autre par le choix de l'utilisateur sur la densité de points par profil à conserver, par la vitesse du balayage de la caméra effectué par l'utilisateur et aussi par l'algorithme de maillage des surfaces qui équilibre la densité ou l'espacement des points dans les axes du profil et du balayage. Bref la dimension moyenne des triangles des surfaces d'entrée ainsi obtenue devrait être conservée dans la surface de sortie pour avoir le même niveau de détails et de précision que celui désiré lors de l'acquisition des surfaces. Il est donc inutile d'utiliser une résolution de grille trop élevée qui générerait des triangles plus petits qu'en entrée car les détails ne seront pas pertinents et ils seront plutôt considérés comme une source de bruit. À l'inverse, une grille de résolution trop grossière résulterait en une perte d'information par rapport aux surfaces d'entrée. La résolution à utiliser est donc celle qui générerait des triangles dans la surface de sortie de même dimension en moyenne que ceux dans les surfaces en entrée.

Une méthode de calcul automatisée de la résolution de la grille a donc été élaborée. Une surface quelconque et réelle a été utilisée pour faire des essais. Plusieurs fonctions de champ avec des grilles de résolutions variables de 1 à 5 millimètres par saut de 0,5 millimètre ont été calculées à partir de cette surface. Cette plage de résolutions a été sélectionnée expérimentalement comme étant la plage la plus probable dans laquelle la grandeur des triangles générés serait de dimension similaire aux triangles de la plupart des surfaces d'entrée. Ensuite les fonctions de champ ont été triangulées pour revenir à

des surfaces explicites dans tous les cas. La surface test contient une seule surface et aucune intégration ou fusion de surface n'est réalisée. Il s'agit simplement de partir d'une surface explicite unique, de calculer sa fonction de champ pour la représenter dans le domaine implicite et finalement de trianguler cette fonction de champ pour revenir à une surface équivalente à celle de départ. Évidemment le résultat est équivalent à la surface de départ à l'intérieur d'un certain seuil de tolérance d'erreur mais les surfaces ne sont pas identiques et ne contiennent pas les mêmes triangles. Cet aspect d'erreur sur les surfaces sera traité dans l'analyse des résultats. Les aires et les périmètres moyens des triangles des surfaces de sortie ont été calculés et deux graphiques ont été créés. Celui de la résolution en fonction de l'aire moyenne et celui de la résolution en fonction du périmètre moyen. La relation entre le périmètre moyen des triangles générés et la résolution de la grille utilisée est parfaitement linéaire dans ce cas-ci. La figure 2.6 présente un graphique de la résolution de la grille en fonction du périmètre moyen des triangles. L'équation ainsi que le coefficient de corrélation de la courbe de tendance sont présentés à la figure 2.6.

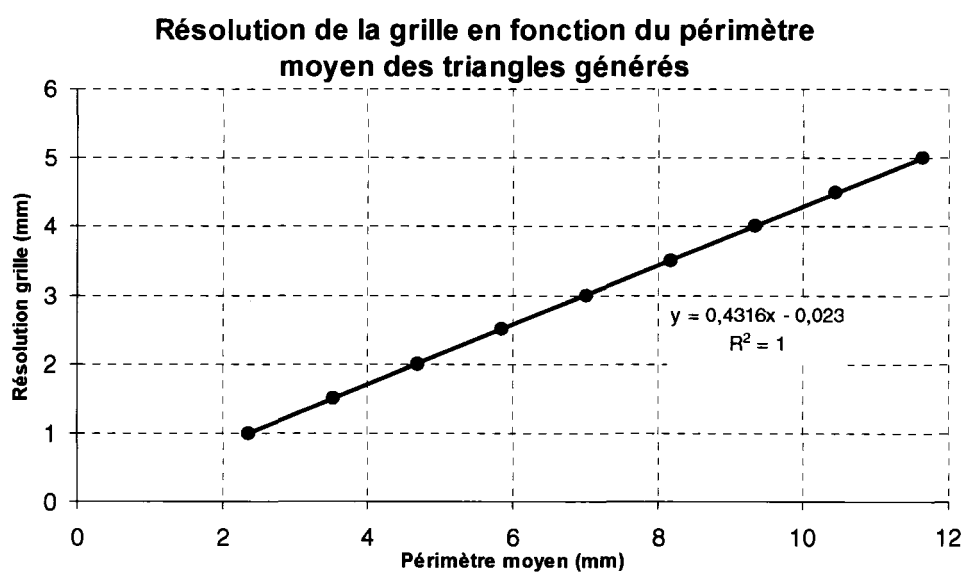


Figure 2.6 Graphique de la résolution en fonction du périmètre moyen

La première méthode de validation de l'équation de la courbe du graphique à la figure 2.6 a consisté à calculer le périmètre moyen des triangles de la même surface de test en entrée et à utiliser cette valeur dans l'équation pour déterminer la résolution adéquate pour cette surface. La fonction de champ a été calculée avec cette résolution et le maillage de la fonction de champ résultante a donné une surface finale. Le périmètre moyen des triangles de cette surface a été calculé et comparé à celui de la surface d'entrée. Les résultats sont pratiquement identiques, c'est-à-dire avec une erreur de l'ordre de 0,1%.

La seconde méthode de validation de l'équation a consisté à tester avec plusieurs autres surfaces. Le processus de test se déroule selon les étapes suivantes :

1. Le périmètre moyen des triangles des surfaces initiales est calculé.
2. L'équation est utilisée pour déterminer la résolution en fonction du périmètre moyen calculé.
3. La fonction de champ est calculée avec la résolution déterminée.
4. La fonction de champ calculée est triangulée pour obtenir une surface polygonale.
5. Le périmètre moyen des triangles de la surface polygonale résultante est calculé.
6. Le périmètre moyen calculé à l'étape 5 est comparé à celui calculé à l'étape 1.

Dans tous les cas, les périmètres comparés à l'étape 6 sont très similaires et la plus grande erreur rencontrée a été de 6,7%. Ces résultats sont largement satisfaisants pour l'application. Le but est de générer des surfaces finales ayant des triangles de dimensions similaires aux surfaces d'entrée afin de conserver le niveau de détails des surfaces sans toutefois introduire du bruit supplémentaire. L'équation 2.1 provenant de la courbe du graphique à la figure 2.6 est donc utilisée pour déterminer de façon automatique la résolution de la grille de la fonction de champ pour obtenir une surface de sortie avec des triangles de même dimension que les surfaces en entrée.

$$r = 0,4316p - 0,023 \quad (2.1)$$

La variable indépendante p représente le périmètre moyen des triangles de la surface d'entrée. Normalement il y a plus d'une surface en entrée et le périmètre moyen calculé est celui de l'ensemble des surfaces. La variable dépendante r représente la résolution à utiliser pour la grille de la fonction de champ.

2.1.2 Recherche du point le plus près sur la surface

Dans le calcul de la fonction de champ pour une surface, la partie la plus importante est la recherche du point le plus près sur la surface pour chaque élément de la grille. C'est l'étape la plus complexe en terme d'algorithme et la plus longue à exécuter. Elle prend plus de 99% du temps de calcul de la fonction de champ. Une fois que ce point est déterminé, les autres calculs sont assez simples. Cette étape est longue et complexe étant donné la nature des surfaces polygonales composées de triangles. Pour un élément de la grille en particulier, il faut trouver le point le plus près sur chacun des triangles de la surface et conserver le plus près entre tous les points. L'algorithme qui détermine le point le plus près d'un élément sur un triangle doit donc être répété plusieurs milliers de fois selon le nombre de triangles de la surface. Plusieurs algorithmes peuvent être utilisés pour déterminer le point le plus près sur une surface triangulée. Certains ont été élaborés et testés sur des surfaces tests. La méthode la plus efficace en terme de temps de calcul sur l'ordinateur a été conservée.

La méthode débute par le calcul de la projection orthogonale du point de l'élément sur le plan d'un triangle. Le point de l'élément qui est projeté est toujours le coin inférieur dans les trois axes de l'élément. La figure 2.7 illustre la projection d'un point sur le plan d'un triangle ainsi que les différentes régions possibles où le point projeté peut se situer.

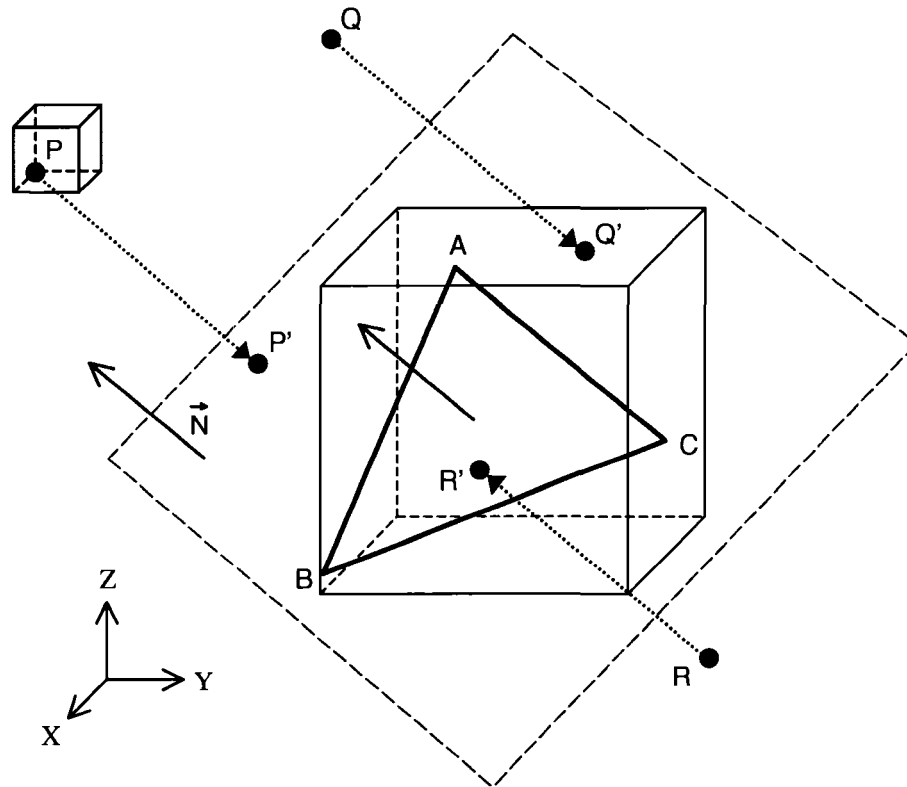


Figure 2.7 Projection orthogonale d'un point sur un plan en 3D

Le triangle à la figure 2.7 est défini par ses sommets A , B et C . La boîte de contour du triangle est illustrée en vert et le plan du triangle est représenté par le rectangle en traits pointillés bleus. Le vecteur N^1 qui est normal au triangle et au plan est identifié par les flèches rouges. Les points P , Q et R représentent des éléments de la fonction de champ et les points P' , Q' et R' représentent les projections orthogonales de ces points sur le plan du triangle. Du point de vue à la figure 2.7, les points P et Q sont au-dessus du plan du triangle alors que le point R est en dessous. L'élément discret P est dessiné pour illustrer la projection à partir du coin inférieur de l'élément. La projection d'un point sur le plan d'un triangle peut se situer à l'intérieur de l'une de trois régions distinctes sur le plan. Elle peut se situer à l'intérieur du triangle (projection R'), elle peut se situer à

¹ La notation en caractère gras dans le texte indique un vecteur.

l'extérieur du triangle mais à l'intérieur de la partie du plan qui est délimitée par la boîte de contour du triangle (projection Q') et finalement elle peut se situer à l'extérieur de la partie du plan qui est délimitée par la boîte de contour (projection P'). Cette dernière projection est nécessairement aussi à l'extérieur du triangle. L'équation 2.2 est utilisée pour calculer les coordonnées d'une projection sur un plan à partir d'un point de l'espace.

$$\vec{P}' = \vec{P} + \vec{N} \left((\vec{N} \bullet \vec{A}) - (\vec{N} \bullet \vec{P}) \right) \quad (2.2)$$

Les variables utilisées à l'équation 2.2 font référence à celles de la figure 2.7. Le vecteur P' représente les coordonnées (x,y,z) de la projection du point P sur le plan dont le vecteur normal est le vecteur N . Les vecteurs A et P sont simplement des vecteurs qui partent de l'origine et qui vont aux points A et P . Le point A doit être un point sur le plan de projection et dans le contexte actuel, c'est aussi un des sommets du triangle. L'opérateur « \bullet » représente un produit scalaire entre deux vecteurs. Le produit scalaire entre deux vecteurs en 3D est défini par l'équation 2.3.

$$\vec{U} \bullet \vec{V} = U_x V_x + U_y V_y + U_z V_z = |\vec{U}| |\vec{V}| \cos(\alpha) \quad (2.3)$$

Le produit scalaire entre deux vecteurs U et V est défini par la multiplication des modules des vecteurs et du cosinus de l'angle α entre eux-ci. Le module d'un vecteur est défini par l'équation 2.4.

$$|\vec{U}| = \sqrt{U_x^2 + U_y^2 + U_z^2} \quad (2.4)$$

Une première vérification est faite pour déterminer si le point projeté est à l'intérieur ou non de la boîte de contour du triangle. L'idée de cette vérification est de déterminer rapidement si le point est à l'intérieur ou non du triangle en comparant simplement les coordonnées du point avec celles des sommets de la boîte. Si le point est à l'extérieur de

la boîte, il est nécessairement à l'extérieur du triangle et ce cas est le plus fréquent. Si le point est à l'intérieur de la boîte de contour alors un calcul est fait pour déterminer si le point est à l'intérieur du triangle ou non. Pour effectuer ce calcul, il faut définir trois vecteurs qui partent du point projeté sur le plan du triangle et qui vont vers les trois sommets du triangle. La figure 2.8 illustre ces vecteurs pour les deux cas possibles.

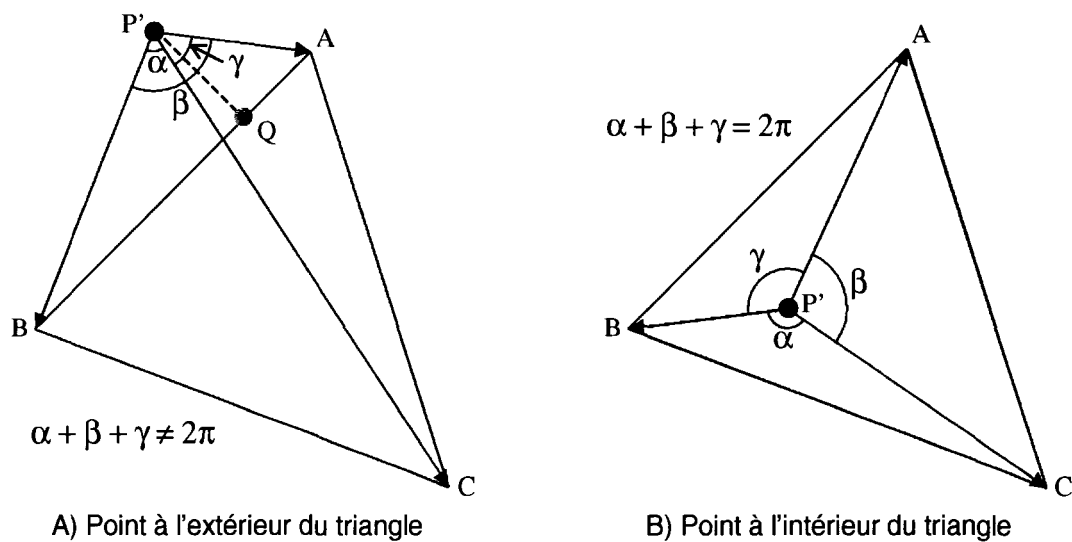


Figure 2.8 Vecteurs du point projeté aux sommets du triangle

À la figure 2.8, le point P' représente le point projeté sur le plan du triangle. Les vecteurs sont définis de ce point vers les trois sommets A , B et C du triangle. Ensuite il faut calculer la somme des angles α , β , et γ entre ces trois vecteurs. Si la somme donne 360° , alors le point est à l'intérieur du triangle (figure 2.8B), sinon, il est à l'extérieur (figure 2.8A). Si le point est à l'intérieur, le point le plus près sur le triangle est sélectionné puisqu'il s'agit du point P' qui correspond à la projection orthogonale du point original P . Dans ce cas, aucun autre point sur le triangle ne sera plus près du point original P . Si le point projeté a été déterminé à l'extérieur du triangle, soit par la vérification rapide de la boîte de contour ou par calcul, alors le point le plus près est nécessairement sur une des arêtes du triangle. Ce point est le point Q pour le cas illustré

à la figure 2.8A. La procédure se poursuit pour évaluer le point le plus près sur le triangle.

À ce stade de l'algorithme, le point le plus près de celui de l'élément original sur le plan du triangle a été déterminé à l'extérieur du triangle. Il faut maintenant poursuivre la recherche en évaluant le point le plus près de l'élément sur une arête du triangle. Ce calcul est fait pour les trois arêtes du triangle et le point le plus près des trois points trouvés est conservé comme étant celui le plus près sur le triangle. Pour déterminer le point le plus près sur une arête d'un triangle, une autre projection orthogonale du point de l'élément sur la droite qui passe par l'arête du triangle est effectuée. Si la projection se situe sur l'arête du triangle alors le point projeté est celui qui est le point le plus près pour cette arête du triangle. Sinon c'est une des deux extrémités de l'arête, soit celle la plus près de la projection. La figure 2.9 illustre cette nouvelle projection et les différents cas possibles.

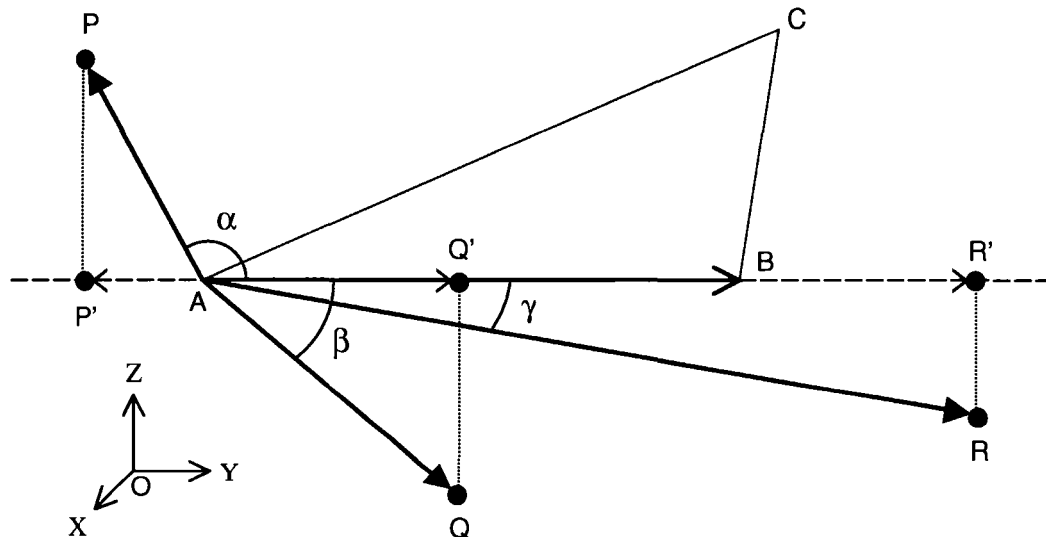


Figure 2.9 Projection d'un point sur une droite en 3D

À la figure 2.9, les projections de trois points P , Q et R sont réalisées sur la droite qui passe par l'arête AB du triangle ABC . Les points projetés sont respectivement P' , Q' et R' . Dans le cas du point P , la projection P' n'arrive pas sur l'arête AB du triangle et le point le plus près du point P sur cette arête est donc le sommet A du triangle. Ce cas se présente si l'inéquation 2.5 est vérifiée.

$$\overrightarrow{AB} \bullet \overrightarrow{AP} \leq 0 \quad (2.5)$$

Si l'inéquation 2.5 est vérifiée, ceci signifie que l'angle α entre les vecteurs \overrightarrow{AB}^1 et \overrightarrow{AP} est droit ou obtus et que le sommet A du triangle est le point recherché. Dans le cas du point R , la projection R' n'arrive toujours pas sur l'arête AB du triangle et le point le plus près du point R sur cette arête est donc le sommet B du triangle. Ce cas se présente si l'inéquation 2.6 est vérifiée.

$$\frac{\overrightarrow{AB} \bullet \overrightarrow{AR}}{|\overrightarrow{AB}|} = |\overrightarrow{AR}| \cos(\gamma) = |\overrightarrow{AR'}| \geq |\overrightarrow{AB}| \quad (2.6)$$

Si l'inéquation 2.6 est vérifiée, ceci signifie que la grandeur du vecteur $\overrightarrow{AR'}$ est plus grande que celle du vecteur \overrightarrow{AB} et que le sommet B du triangle est le point recherché. Dans le cas du point Q , la projection Q' arrive sur l'arête AB du triangle et le point le plus près du point Q sur l'arête AB est le point projeté Q' . Ce cas se présente si les inéquations 2.5 et 2.6 ne sont pas vérifiées. L'équation 2.7 est utilisée pour déterminer les coordonnées du point Q' .

$$\vec{Q'} = \vec{A} + \left(\frac{\overrightarrow{AB}}{|\overrightarrow{AB}|} \right) \left(\frac{\overrightarrow{AB} \bullet \overrightarrow{AQ}}{|\overrightarrow{AB}|} \right) \quad (2.7)$$

¹ Le vecteur \overrightarrow{AB} est un vecteur qui part du point A et qui arrive au point B .

Les coordonnées du vecteur \mathbf{Q}' , qui part de l'origine et qui se rend au point Q' , calculées à l'aide de l'équation 2.7 sont les mêmes que celles du point Q' recherché. Une fois le point le plus près sur l'arête déterminé, il est conservé et la procédure se répète pour les deux autres arêtes du triangle. Les trois distances entre le point de l'élément et les trois points trouvés sur chacune des arêtes sont calculées et le point le plus près est conservé. Il s'agit du point le plus près du point de l'élément sur le triangle. La distance d entre deux points A et B de l'espace est calculée selon l'équation 2.8.

$$d = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2 + (A_z - B_z)^2} \quad (2.8)$$

La distance calculée à l'équation 2.8 représente la distance euclidienne entre deux points. Une fois le processus terminé et que le point le plus près a été trouvé pour un triangle, il suffit de le répéter pour tous les triangles afin de trouver le plus près de tous. La figure 2.10 résume la procédure de la recherche du point le plus près d'un élément sur un triangle à l'aide d'un ordinogramme.

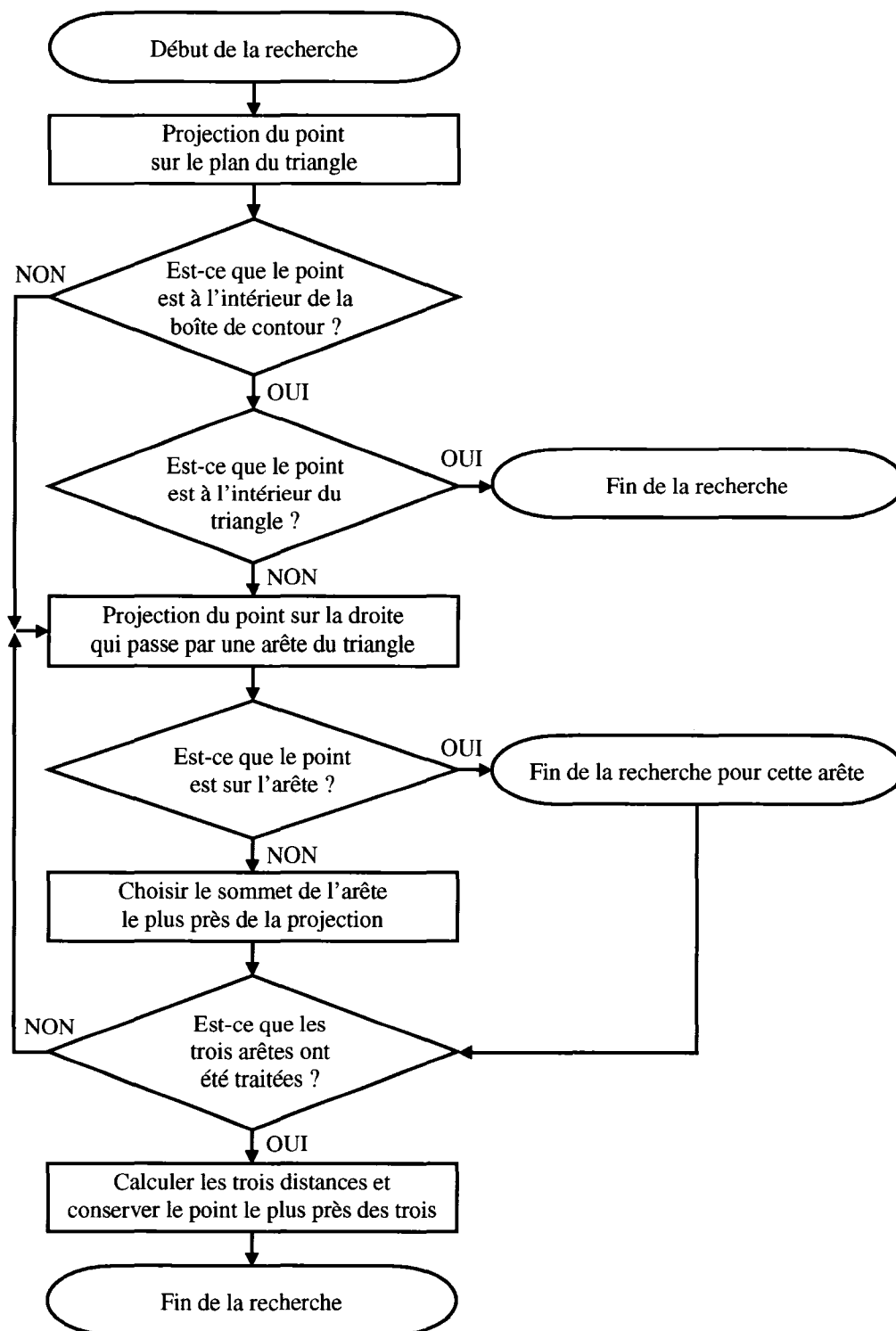


Figure 2.10 Procédure de recherche du point le plus près sur un triangle

2.1.3 Optimisations du temps de calcul

Le calcul des fonctions de champ est la première de trois étapes dans le processus de fusion des surfaces et c'est aussi la plus longue en temps de calcul à cause de la recherche du point le plus près tel que mentionné précédemment. Deux optimisations distinctes ont été développées pour diminuer le temps de calcul des fonctions de champ. Ces optimisations sont présentées et décrites dans les sections suivantes.

2.1.3.1 Éléments utiles

La première méthode d'optimisation est basée sur le principe des éléments utiles. Pour une surface quelconque, il n'est pas nécessaire d'évaluer la fonction de champ à tous les éléments dans la boîte de contour de la surface. Le but de calculer la fonction de champ est d'être en mesure de fusionner les surfaces dans ce domaine implicite. Le processus de fusion utilise seulement les éléments qui sont sur la surface ou près de celle-ci. Les éléments près de la surface sont ceux qui se trouvent à l'intérieur de la distance de redondance établie. Cette distance dépend de la précision du système de positionnement et elle sera détaillée dans le chapitre suivant qui traite de l'intégration des surfaces. La figure 2.11 présente un exemple graphique des éléments utiles d'une surface.

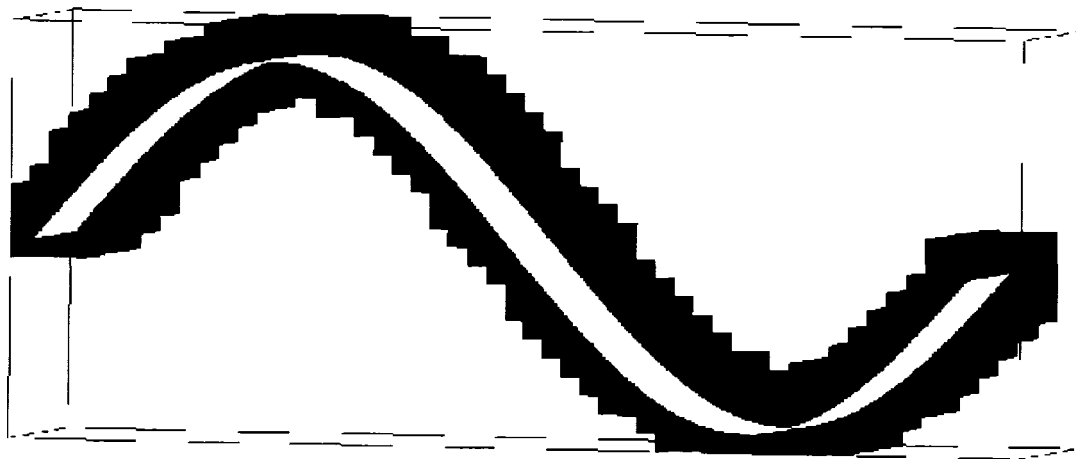


Figure 2.11 Éléments utiles d'une fonction de champ

À la figure 2.11, la surface dans l'espace 3D est une surface de synthèse de forme sinusoïdale. La fonction de champ est définie à l'intérieur de la boîte de contour qui est représentée à la figure 2.11. Seulement les éléments utiles de la fonction de champ sont affichés. Certains éléments sont en jaune et d'autres en gris pour distinguer les éléments de contour des autres. La fonction de champ sera évaluée seulement à ces éléments utiles. Les éléments utiles d'une fonction de champ représentent de 10% à 15% du volume total compris dans la boîte de contour pour différentes surfaces réelles. En évaluant seulement ces éléments utiles, le temps de calcul de la fonction de champ est pratiquement diminué d'un facteur dix.

2.1.3.2 Plan de projection des triangles

La seconde méthode d'optimisation est basée sur un plan de projection des triangles de la surface explicite de départ pour accélérer la recherche du point le plus près. Comme cette recherche est excessivement longue, une attention particulière a été portée pour l'optimiser. L'idée est de limiter le nombre de triangles de la recherche. Pour un élément donné, plutôt que de rechercher le point le plus près sur tous les triangles de la surface et conserver le plus près de tous, ce processus est fait seulement sur quelques triangles de la surface qui ont été judicieusement sélectionnés. La façon de sélectionner ces triangles est fiable et elle donne toujours le bon résultat en comparant avec la recherche exhaustive sur tous les triangles.

Au départ, un des trois plans du système de coordonnées est choisi pour servir de plan de projection des triangles. Ce plan de projection est borné par les limites d'un des côtés de la boîte de contour de la surface. Le plan est divisé par une grille de même résolution que celle de la fonction de champ. Les deux grilles sont alors alignées. La figure 2.12 illustre les trois plans de projection possibles.

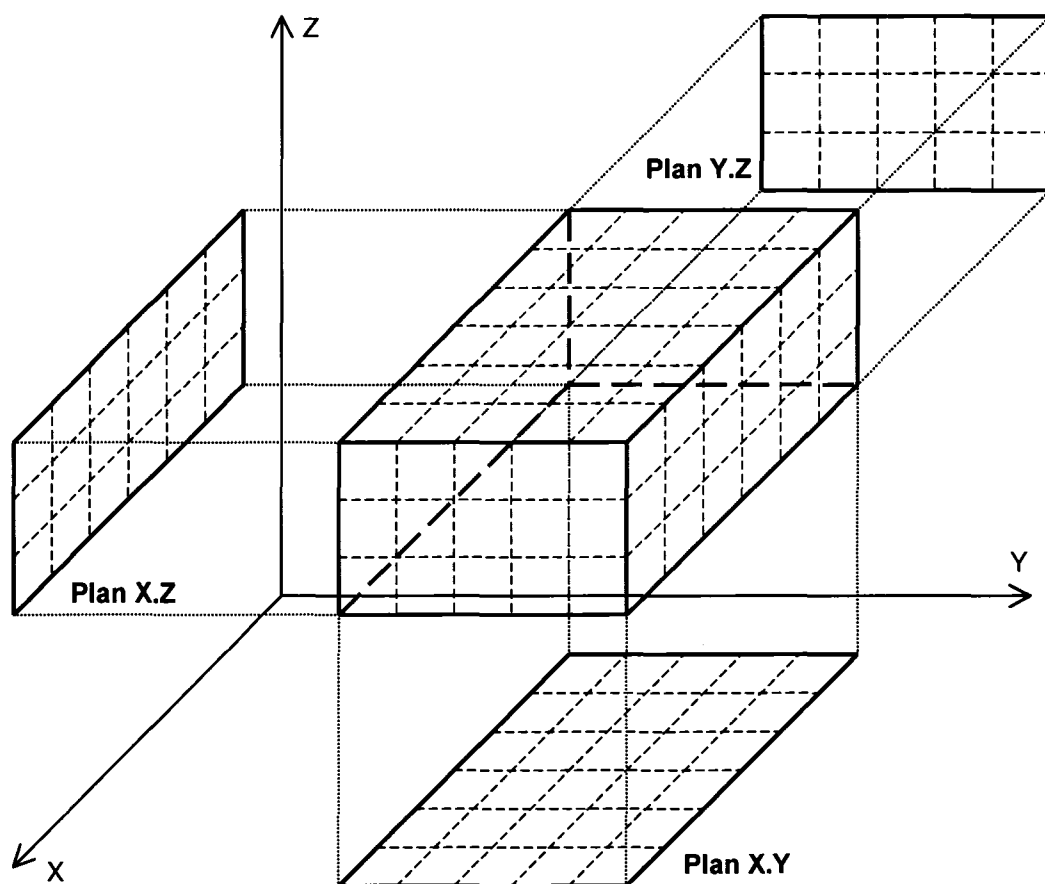


Figure 2.12 Différents plans de projection des triangles

La boîte de contour de la surface est représentée en bleu à la figure 2.12. Les trois plans de projection possibles sont dessinés en rouge. Les quadrillages en traits pointillés illustrent les éléments de la fonction de champ dans le cas de la boîte de contour ainsi que les éléments des plans de projection. La sélection du plan parmi les trois possibles est faite selon un critère de dispersion maximale des triangles de la surface dans le plan de projection. Les vecteurs normaux à tous les triangles sont normalisés et additionnés et le plan choisi est celui dont le vecteur normal correspond à l'axe du système de coordonnées qui a la composante maximale dans la somme des vecteurs normaux des triangles. Par exemple, si la somme des vecteurs normaux de tous les triangles donne le vecteur $(3,4,5)$, alors la composante maximale est en Z et l'axe Z est le vecteur normal

au plan (X,Y) qui est sélectionné comme plan de projection. Ce critère de sélection du plan de projection minimise le nombre de triangles qui seront traités lors de la recherche du point le plus près. Chacun des triangles de la surface est projeté sur le plan avec une projection orthogonale. Pour un triangle donné, tous les éléments de la grille du plan de projection qui touchent à la projection du triangle sont étiquetés avec l'indice du triangle. La figure 2.13 présente la projection d'un triangle sur le plan de projection des triangles.

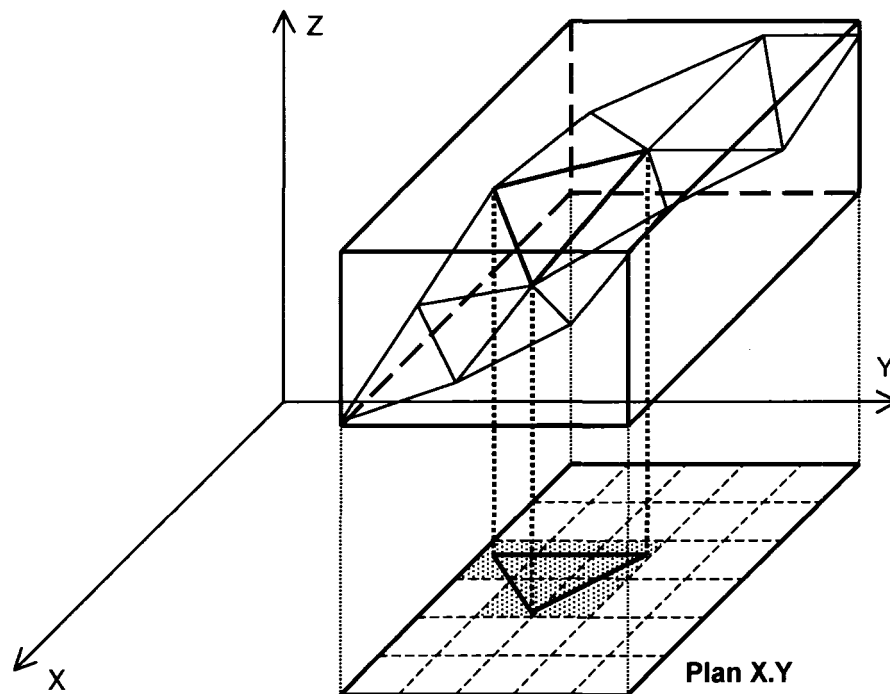


Figure 2.13 Projection d'un triangle sur le plan de projection

À la figure 2.13, le triangle de la surface qui est en vert est projeté sur le plan (X,Y) qui représente le plan de projection choisi. Les cinq éléments ombragés du plan de projection sont étiquetés avec l'indice de ce triangle. Suite à la projection de tous les triangles, chacun des éléments du plan peut contenir zéro, un ou plusieurs étiquettes de triangles.

La recherche du point le plus près pour un élément donné de la fonction de champ s'effectue de la façon suivante. L'élément en question est projeté sur le plan de projection. Cet élément coïncide exactement avec un élément du plan de projection puisque les grilles sont alignées. Cet élément atteint du plan de projection des triangles contient les indices des triangles qui y touchaient lors de leur projection. Puisque l'élément de la fonction de champ et les triangles dont les indices sont contenus dans l'élément atteint du plan de projection ont été projetés au même endroit, la recherche du point le plus près est effectuée sur ces triangles seulement dans un premier temps. Un premier résultat potentiel est donc obtenu si l'élément du plan de projection contient l'indice d'un triangle ou plus.

Ensuite la recherche se poursuit sur les éléments immédiatement voisins de l'élément initial de projection. Seulement les triangles dont les indices se trouvent dans ces éléments voisins sont traités. Et ainsi de suite avec les éléments voisins des premiers voisins toujours en s'éloignant de l'élément initial. Si l'indice d'un triangle se retrouve dans plusieurs éléments voisins, il sera traité seulement une fois grâce à un système efficace d'étiquetage des triangles traités. Cette recherche se termine en vérifiant un critère d'arrêt. Lorsque la distance réelle en 3D entre l'élément de la fonction de champ et le point le plus près trouvé jusqu'à présent sur un triangle est plus petite que la distance en 2D dans le plan de projection entre la projection de l'élément de la fonction de champ et les prochains éléments voisins à traiter du plan de projection, la recherche s'arrête. Le critère d'arrêt est basé sur le fait que la distance réelle 3D entre l'élément de la fonction de champ et les triangles contenus dans les prochains éléments voisins à traiter du plan de projection sera égale ou plus grande que leur distance 2D dans le plan de projection. La figure 2.14 illustre le principe du plan de projection et la séquence de recherche des triangles.

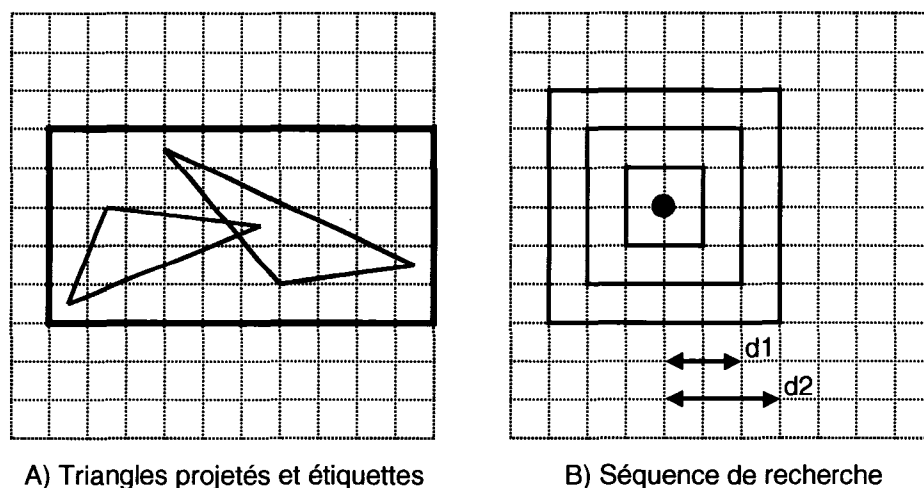


Figure 2.14 Plan de projection des triangles et séquence de recherche

Aux figures 2.14A et 2.14B, les grilles représentent le plan de projection. À la figure 2.14A, la projection orthogonale dans le plan pour deux triangles d'une surface quelconque est illustrée. Il est à noter que dans ce cas particulier les triangles de la surface auraient nécessairement une profondeur et une orientation différentes dans la troisième dimension qui n'est pas illustrée par le plan de projection. Ceci pour éviter que les triangles de la surface se croisent réellement en 3D comme les projections 2D de ces triangles le laissent sous-entendre étant donné la perte d'information sur la troisième dimension. La définition du format des surfaces ne supporte pas que des triangles de la surface se croisent et s'entrecoupent réellement en 3D. Les éléments en bleu sont étiquetés seulement avec l'indice du triangle dont la projection est dessinée en bleu. La même procédure est appliquée à la projection et aux éléments en vert. Les éléments en jaune sont étiquetés avec les indices des deux triangles dont les projections sont dessinées. Les autres éléments en blanc n'auraient pas d'indice de triangles si la surface était uniquement composée de ces deux triangles. Dans ce dernier cas, le plan de projection serait borné par le rectangle noir avec la ligne en caractère gras.

La figure 2.14B illustre la séquence de recherche pour un élément en particulier. Le point noir indique la projection d'un élément de la fonction de champ sur un élément du

plan. La recherche commence par les triangles dont les indices sont contenus dans cet élément. Ensuite la recherche se poursuit avec les triangles dont les indices sont contenus dans les éléments sur le carré rouge. Et ainsi de suite avec le carré bleu suivi du carré vert en vérifiant le critère d'arrêt à chacun des niveaux de nouveaux voisins à considérer. Par exemple, si l'algorithme de recherche a terminé de traiter les éléments sur le carré rouge, alors le critère d'arrêt est vérifié avant de passer aux éléments sur le carré bleu. La distance 3D réelle est calculée entre l'élément de la fonction de champ et le point le plus près trouvé jusqu'à présent sur les triangles dont les indices sont contenus dans l'élément du plan de projection où l'élément de la fonction de champ a été projeté ainsi que dans les éléments du carré rouge. Si cette distance est plus grande que la distance $d1$ à la figure 2.14B, alors la recherche se poursuit sur les nouveaux triangles dont les indices sont contenus dans les éléments du carré bleu. La distance $d1$ représente la distance minimale possible en 3D entre l'élément de la fonction de champ et un point sur un triangle dont l'indice est contenu dans un élément sur le carré bleu. Cette affirmation est valide à condition que l'indice du triangle ne soit pas contenu dans les éléments du carré rouge ou dans l'élément initial de recherche. Il est possible qu'un point sur ces nouveaux triangles soit plus près que le point le plus près actuel. Suite au traitement des éléments sur le carré bleu, le critère d'arrêt est vérifié à nouveau. La distance 3D réelle est calculée entre l'élément de la fonction de champ et le point le plus près actualisé sur tous les triangles incluant ceux dont les indices sont contenus dans les éléments du carré bleu. Si cette distance est plus petite que la distance $d2$ à la figure 2.14B, alors la recherche s'arrête. Il est impossible qu'un point sur un nouveau triangle dont l'indice est contenu dans les éléments du carré vert soit plus près que le point le plus près actuel.

Cette phase d'optimisation à elle seule est très performante et elle est encore plus efficace lorsqu'elle est combinée à la première optimisation sur les éléments utiles. Comme les éléments utiles sont près de la surface, les distances 3D trouvées lors de la recherche du point le plus près sont relativement petites et le critère d'arrêt de la

recherche dans le plan de projection est atteint assez rapidement. Par exemple, en combinant les deux types d'optimisation, environ cinq triangles ont été traités sur une surface d'environ cinq milles triangles. Il y a donc une diminution d'un facteur 1 000 sur le temps de calcul par rapport à la méthode exhaustive de traitement de tous les triangles. Si le facteur 10 de l'autre optimisation est ajouté, le traitement d'une surface peut passer de plusieurs heures à quelques secondes seulement en utilisant les optimisations mises au point pour le calcul de la fonction de champ.

2.1.4 Implémentation informatique

Les sections précédentes de ce chapitre décrivent différents concepts associés à la fonction de champ sans nécessairement spécifier les détails des méthodes employées pour réaliser les calculs. Cette section traite de l'implémentation informatique du calcul de la fonction de champ. Plusieurs classes (ensemble de programmes en C++) d'usage général et de manipulation d'objets 3D ont été développées à l'INO et sont rassemblées dans une bibliothèque. Certaines de ces classes ont été développées spécifiquement pour le projet du MapScan et pour le projet de fusion des surfaces.

Les données d'entrée du processus de fusion sont des surfaces polygonales dans l'espace 3D. Une classe qui s'appelle « TriangleMesh » contient toutes les structures de données nécessaires pour supporter une surface dans l'espace 3D. Cette classe contient entre autres trois listes. Une liste de triangle, une liste d'arêtes et une liste de sommets qui contiennent des pointeurs sur différents types de données. Par exemple un triangle contient trois pointeurs sur ses trois arêtes. Une arête contient deux pointeurs sur ses deux sommets. Un sommet contient des pointeurs sur toutes les arêtes qui y sont reliées. Une arête contient aussi un ou deux pointeurs sur le ou les deux triangles qui y sont reliés selon que l'arête délimite ou non le contour de la surface. Un sommet contient trois valeurs réelles à point flottant qui correspondent à ses coordonnées dans l'espace. Un triangle contient un pointeur sur un vecteur qui est normal à la surface du triangle. Plusieurs méthodes d'accès à la surface sont disponibles. Il est donc possible

d'interroger une surface pour obtenir de l'information sur celle-ci. De plus il est possible de modifier la surface en ajoutant ou en enlevant des triangles. Le type de surface supporté est identique au format STL.

Des programmes ont été développés dans le but de calculer la fonction de champ à partir d'une surface polygonale. Une classe qui s'appelle « FieldFunction » a été développée contenant la structure de données nécessaire pour supporter une fonction de champ. Elle contient entre autres un tableau à trois dimensions. Chacun des éléments du tableau représente un élément de la fonction de champ et il contient un pointeur sur une autre structure. Celle-ci contient la valeur réelle à point flottant de la distance à la surface, le vecteur normal à la surface à l'endroit du point le plus près ainsi que l'information booléenne sur l'appartenance ou non au contour de la surface du point le plus près. Le vecteur normal à la surface à l'endroit du point le plus près est calculé en fonction des vecteurs normaux des triangles concernés. Le point le plus près est soit sur la surface d'un triangle sans toucher à son contour, soit sur une arête ou encore c'est un sommet d'un triangle. Selon le cas, ce point appartient à un ou plusieurs triangles simultanément. Le vecteur normal à la surface à ce point est calculé en effectuant la somme des vecteurs normaux de l'ensemble de ces triangles. Un élément peut être utile ou non à sa fonction de champ tel que décrit précédemment. Si l'élément n'est pas utile alors son pointeur pointe sur NULL.

La classe « TriangleMeshPlaneProjection » a été développée et contient la structure de données nécessaire pour supporter le plan de projection de l'optimisation décrite précédemment. Elle contient entre autre un tableau à deux dimensions. Chacun des éléments du tableau représente un élément du plan de projection et il contient un pointeur sur une liste de triangles. Si aucun triangle n'a été projeté sur un élément alors le pointeur de cet élément pointe sur NULL. Les triangles dans la liste contenue dans un élément correspondent évidemment aux triangles qui ont été projetés sur cet élément.

À partir d'une surface polygonale dans l'espace 3D, les étapes suivantes sont réalisées pour en arriver à la fonction de champ. La résolution de la grille de la fonction de champ est calculée automatiquement avec le périmètre moyen des triangles de la surface. La boîte de contour de la surface est calculée. La grille de la fonction de champ est créée à l'intérieur de la boîte de contour en utilisant la résolution calculée. À cette étape les éléments de la grille se trouvent à des coordonnées quelconques dans l'espace. La grille est alors déplacée légèrement dans l'espace pour que les éléments soient alignés sur un multiple entier de sa résolution. La grille est déplacée négativement sur les trois axes jusqu'au multiple entier inférieur de sa résolution. Cette opération est nécessaire pour assurer l'alignement entre toutes les fonctions de champ à intégrer en une seule fonction de champ globale. Si dans une zone donnée de l'espace il y a de la redondance entre plusieurs surfaces qui se superposent, il y aura plusieurs fonctions de champ définies dans cette zone. Suite à l'alignement des grilles, les coordonnées des éléments de toutes les fonctions de champ dans cette zone coïncideront aux même endroits dans l'espace.

Les éléments utiles de la fonction de champ sont déterminés en calculant les boîtes de contour individuelles de tous les triangles de la surface et en sélectionnant tous les éléments qui se trouvent à l'intérieur ou qui touchent à ces boîtes de contour. La boîte de contour de la surface entière qui a servi à délimiter la fonction de champ et les boîtes de contour individuelles des triangles qui ont servi à déterminer les éléments utiles ne sont pas les boîtes de contour réelles de ces objets. Elles ont plutôt été augmentées d'un certain nombre de fois la résolution de la grille dans chacun des axes suite à leur calcul avant d'être utilisées pour les fins de la fonction de champ. Ce nombre n de fois est variable et il dépend de la résolution r de la grille ainsi que de la distance d de redondance établie. L'équation 2.9 est utilisée pour calculer ce paramètre.

$$n = \frac{d}{r} \quad (2.9)$$

Le nombre n de fois la résolution de la grille à l'équation 2.9 est un nombre entier qui est arrondi à l'entier supérieur du résultat du calcul. La procédure qui consiste à augmenter les boîtes de contour est essentielle pour fusionner des surfaces redondantes selon la distance de redondance établie.

Le plan de projection des triangles est déterminé selon le critère de dispersion maximale et il est créé avec un des trois côtés de la boîte de contour de la surface comme domaine ainsi qu'avec la même résolution que la grille de la fonction de champ. Les boîtes de contour individuelles des triangles sont calculées. Il s'agit des boîtes de contour réelles sans qu'elles soient augmentées. Pour un triangle donné, sa boîte de contour est projetée sur le plan en laissant tomber la troisième dimension. Tous les éléments du plan de projection qui se trouvent à l'intérieur ou qui touchent à la boîte de contour reçoivent l'indice de ce triangle comme étiquette. Le processus est répété pour tous les triangles. Cette méthode de projection des boîtes de contour pour compléter le plan n'est pas exacte. Il se peut qu'un élément contienne l'indice d'un triangle qui ne lui touche pas et que quelques triangles soient traités inutilement lors de la recherche du point le plus près. Cependant la méthode est beaucoup plus simple et rapide qu'une autre qui utiliserait la géométrie exacte des triangles.

Le calcul de la fonction de champ est fait seulement sur les éléments utiles. Pour chacun des éléments utiles, le point le plus près sur la surface est déterminé. La valeur de la fonction de champ est la distance entre l'élément et le point le plus près sur la surface. Le vecteur normal à la surface à l'endroit du point le plus près est calculé et conservé avec la valeur de la fonction de champ. L'information de contour est aussi conservée. Les étapes du calcul d'une fonction de champ à partir d'une surface polygonale se résument de la façon suivante :

1. Calculer le périmètre moyen des triangles de la surface.
2. Calculer la résolution de la grille.
3. Calculer la boîte de contour de la surface.

4. Créer la grille de la fonction de champ à l'intérieur de la boîte de contour.
5. Aligner la grille sur un multiple de la résolution.
6. Calculer les boîtes de contour individuelles des triangles.
7. Sélectionner les éléments utiles de la grille.
8. Déterminer et créer le plan de projection.
9. Compléter le plan de projection en y projetant tous les triangles.
10. Pour tous les éléments utiles, déterminer le point le plus près sur la surface.
11. Calculer la distance minimale à la surface.
12. Calculer le vecteur normal à l'endroit du point le plus près.
13. Déterminer si le point le plus près est sur le contour ou non de la surface.

Tout le processus est répété pour chacune des surfaces partielles de l'objet. Une surface partielle est le résultat d'un balayage avec le profilomètre. Un nuage de points a été généré à partir des profils du balayage et il a été maillé pour produire une surface partielle. Il y a donc autant de fonctions de champ qu'il y a de surfaces partielles. Toutes les fonctions de champ ont la même résolution qui est calculée en tenant compte du périmètre moyen des triangles de toutes les surfaces partielles. Les fonctions de champ ainsi calculées sont les données d'entrée pour la prochaine étape, soit l'intégration des fonctions de champ en une seule fonction de champ globale.

2.1.5 Opérations dans le domaine implicite

Lorsque la fonction de champ d'une surface polygonale est calculée, il est possible de réaliser différentes opérations dans ce domaine implicite de représentation, produisant un effet sur la surface finale. D'autres opérations intéressantes, en plus de l'intégration des fonctions de champ, sont possibles et elles laissent entrevoir un très bon potentiel pour la représentation implicite. Différents essais ont été réalisés et dans l'ensemble, il semble que la plupart des opérations possibles sur une image 2D qui sont basées sur un noyau de convolution sont aussi possibles sur une surface dans l'espace 3D. Il suffit de calculer la fonction de champ d'une surface, réaliser l'opération dans le domaine

implicite et trianguler la fonction de champ résultante qui a été modifiée par l'opération pour observer le résultat sur la nouvelle surface. Comme la fonction de champ est une grille 3D, il suffit de définir un noyau de convolution 3D d'une certaine dimension et d'associer des poids à chacun de ces éléments pour en faire un filtre. Un essai particulier a été fait en testant un filtre passe-bas de type moyenneur avec un noyau de trois éléments dans chacun des axes. La figure 2.15 présente le résultat obtenu.

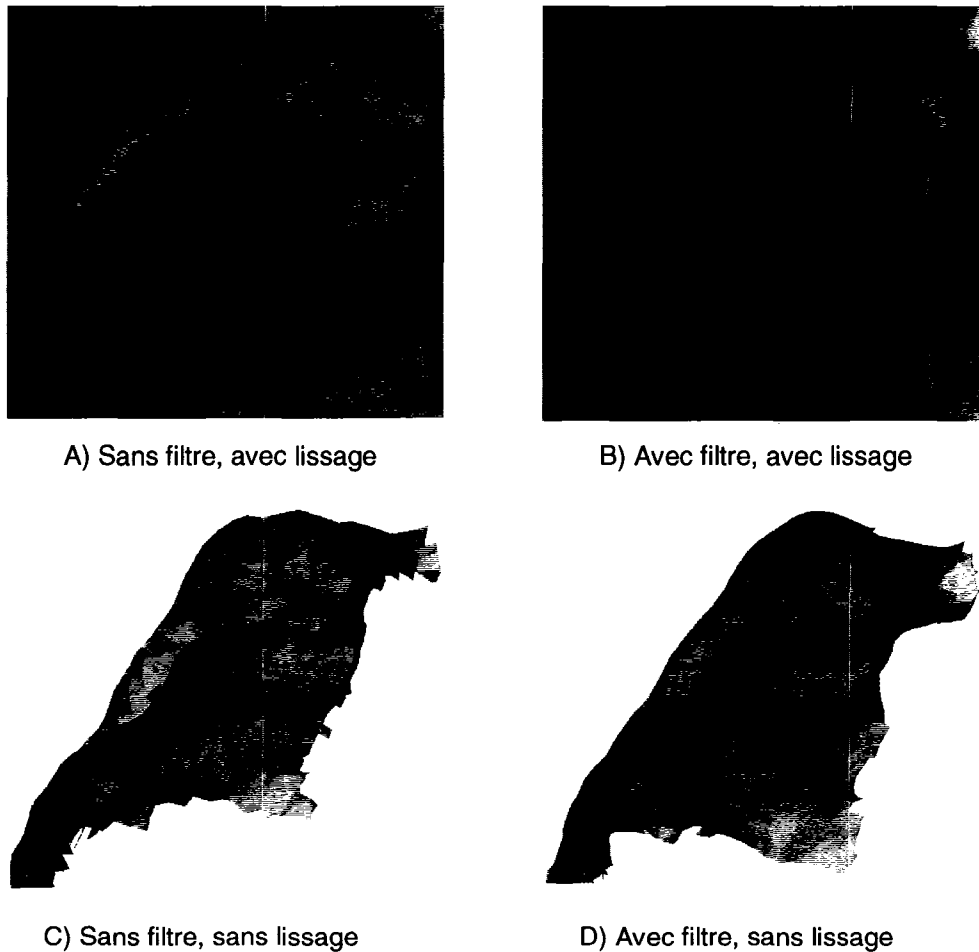


Figure 2.15 Filtrage d'une surface implicite

L'information représentée par les surfaces à la figure 2.15 est la bouche de la statue de Sophocle avec sa moustache et son menton. Aux figures 2.15A et 2.15B les surfaces

sont vues de face alors qu'aux figures 2.15C et 2.15D, elles sont tournées vers la gauche et vers le haut. Toutes les surfaces sont représentées en format solide. Le rendu des surfaces aux figures 2.15A et 2.15B comprend un lissage des triangles qui n'est pas présent sur les surfaces aux figures 2.15C et 2.15D. Ce lissage des triangles n'a rien à voir avec le filtre passe-bas qui est décrit présentement. C'est simplement pour offrir deux rendus différents pour justement mieux apprécier l'effet du filtre passe-bas.

La fonction de champ a été calculée sur la surface originale numérisée. Les figures 2.15A et 2.15C représentent la triangulation de la fonction de champ sans avoir passé le filtre passe-bas sur celle-ci. Les figures 2.15B et 2.15D représentent la triangulation de la fonction de champ qui a été filtrée par le filtre passe-bas. En comparant les figures 2.15A et 2.15B ou encore les figures 2.15C et 2.15D, le filtre rend la surface plus lisse. Le résultat est similaire à celui d'un filtre passe-bas sur une image 2D. Ces opérations dans le domaine implicite peuvent servir à améliorer la qualité des surfaces résultantes de la numérisation d'objets 3D.

2.2 Intégration des fonctions de champ

Cette section présente le principe ainsi que les étapes de l'intégration des fonctions de champ. Ensuite l'implémentation informatique des algorithmes est aussi décrite.

2.2.1 Principe de l'intégration

L'intégration des fonctions de champ représente la fusion de toutes les fonctions de champ individuelles en une seule fonction globale. C'est le cœur du processus de fusion des surfaces. Par exemple, la numérisation de la surface d'un objet 3D qui nécessite deux balayages pour couvrir l'ensemble de sa surface produira deux surfaces polygonales. Les surfaces polygonales doivent se superposer légèrement pour obtenir un résultat final valide. Ces surfaces seront ensuite converties individuellement en deux fonctions de champ distinctes. Les fonctions de champ aussi se superposeront en ayant

des éléments communs puisque les grilles sont alignées et qu'elles sont représentées dans le même référentiel. L'intégration permet d'obtenir une seule fonction de champ globale qui sera définie sur l'ensemble des deux volumes des fonctions de champ individuelles. Cette fonction intégrée représente implicitement la totalité de la surface numérisée par les deux balayages. L'intégration consiste à évaluer de nouvelles valeurs de fonction de champ à chacun des éléments de la fonction de champ globale. Ces nouvelles valeurs de distances à la surface dépendent des valeurs dans les fonctions de champ individuelles. Le processus d'intégration n'est pas limité à deux fonctions de champ. En théorie, une infinité de fonctions de champ pourraient être intégrées en une seule. En pratique, le nombre de fonctions de champ est limité par les ressources informatiques disponibles. La figure 2.16 illustre le principe d'intégration de deux fonctions de champ en 2D.

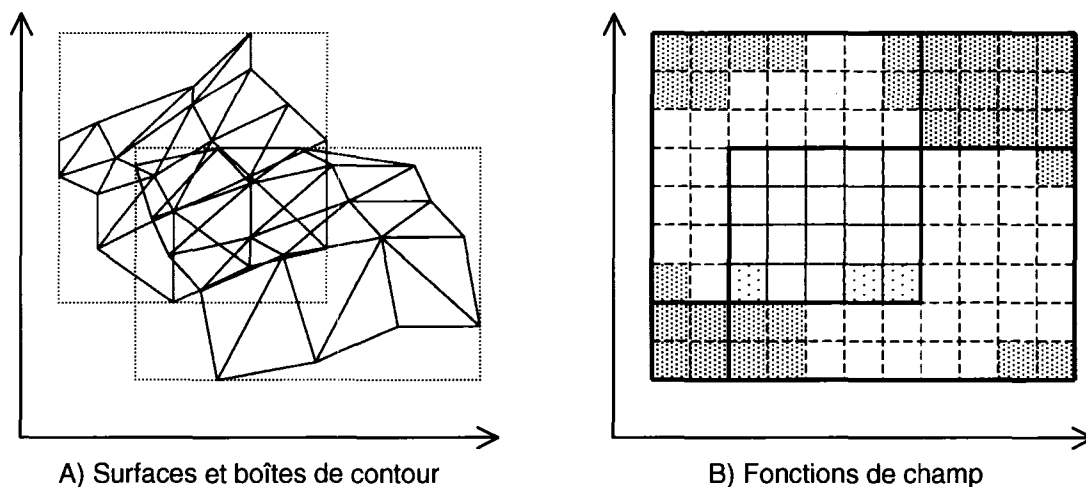


Figure 2.16 Principe d'intégration de deux fonctions de champ

À la figure 2.16A, les deux surfaces polygonales ainsi que leur boîte de contour sont affichées avec des couleurs différentes. À la figure 2.16B, les grilles des fonctions de champ sont affichées. La grille bleue représente la fonction de champ de la surface dessinée en bleu et la grille rouge représente celle de la surface dessinée en rouge à la figure 2.16A. La grille verte représente la fonction de champ intégrée. Les éléments

ombragés représentent des éléments qui ne sont pas utiles à leur fonction de champ. La région commune aux grilles bleues et rouges représente la zone de redondance. La boîte de contour de la fonction de champ intégrée à la figure 2.16B englobe les fonctions de champ individuelles. Les éléments ombragés de toutes les couleurs sont des éléments inutiles à la fonction de champ intégrée à l'exception des trois éléments moins ombragés. Ceux-ci étaient inutiles à leur fonction de champ respective mais deviennent utiles à la fonction de champ intégrée.

2.2.2 Étapes de l'intégration

Le processus d'intégration est divisé en plusieurs étapes. L'intégration se fait un élément à la fois. Toutes les étapes sont répétées pour chacun des éléments de la fonction de champ globale. Dans un premier temps la fonction de champ globale est créée à partir de toutes les fonctions de champ individuelles. La fonction de champ globale possède la même résolution que les autres fonctions de champ et sa grille est aussi alignée avec celles des autres fonctions de façon à ce que les éléments de toutes les fonctions aient les mêmes coordonnées spatiales dans les zones communes. La boîte de contour à l'intérieur de laquelle la fonction de champ globale est définie englobe toutes les boîtes de contour des fonctions de champ individuelles. Tous les éléments de la fonction de champ globale sont parcourus et les étapes suivantes sont effectuées pour chacun d'eux.

2.2.2.1 Détection des éléments utiles

Étant donné que le domaine de la fonction de champ globale englobe toutes les autres fonctions de champ et que ce domaine est une boîte rectangulaire orientée selon les axes du système de coordonnées, il est possible qu'un élément dans l'espace 3D de cette fonction de champ n'appartienne à aucune autre fonction de champ. C'est le cas des éléments ombragés en vert à la figure 2.16B. Comme le domaine d'une fonction de champ est restreint à la boîte de contour de sa surface, ce cas se présente pratiquement

toujours, à la condition qu'au moins deux fonctions aient des limites spatiales différentes sur plus d'un axe. Dans ce cas l'élément est ignoré et aucun calcul n'est fait sur cet élément. Il n'est pas considéré comme un élément utile de la fonction de champ intégrée.

Si un élément appartient à une ou plusieurs fonctions de champ individuelles et que dans tous les cas cet élément n'est pas utile à la fonction selon la méthode d'optimisation des éléments utiles, alors cet élément n'est pas considéré comme utile et il est ainsi ignoré. C'est le cas des éléments ombragés en bleu et en rouge à la figure 2.16B.

Dans le cas où un élément appartient à une ou plusieurs fonctions de champ et qu'il est utile à une seule, ce sont les valeurs de distance et d'information de contour de cette fonction qui sont directement utilisées pour cet élément dans la fonction de champ globale. Ce dernier cas signifie que la surface est relativement près de l'élément et qu'il n'y a pas de redondance à cet endroit, c'est-à-dire qu'il n'y a pas de surfaces superposées. C'est le cas des éléments utiles à leur fonction à la figure 2.16B qui appartiennent seulement à la fonction en bleu ou seulement à la fonction en rouge ainsi que des trois éléments faiblement ombragés dans la zone de redondance.

Si un élément appartient à plusieurs fonctions de champ et qu'il est utile à plus d'une, alors ces éléments utiles sont considérés aux étapes suivantes dans le calcul de la valeur de la fonction de champ intégrée. C'est le cas des éléments non ombragés qui sont utiles aux deux fonctions dans la zone de redondance à la figure 2.16B.

2.2.2.2 Détection des minimums

À partir de tous les éléments utiles à considérer pour un même point de l'espace, les valeurs minimales des fonctions de champ sont détectées afin de trouver les éléments les plus près de la surface, ceux avec les distances minimales en valeur absolue. Deux

éléments minimums sont recherchés, il s'agit de celui avec la valeur minimale dans l'ensemble de tous les éléments de contour ainsi que celui dans les éléments qui ne font pas partie du contour des surfaces.

Si le cas se présente où l'élément minimum de non-contour n'existe pas parce que tous les éléments considérés sont des éléments de contour, alors les valeurs de cet élément dans la fonction de champ intégrée sont directement celles de l'élément minimal de contour trouvé. Si le cas se présente où la distance, en valeur absolue, de l'élément minimal de contour est plus petite que celle de l'élément de non-contour et que la différence entre les deux est plus grande que la distance de redondance, alors les valeurs de cet élément dans la fonction de champ intégrée sont encore celles de l'élément minimal de contour trouvé.

Si aucun des deux cas mentionnés ne s'est présenté, le processus d'intégration se poursuit avec les autres étapes. La distance de redondance sera définie ultérieurement à la section 2.2.3. Si un des deux cas mentionnés est rencontré, ceci signifie que l'élément le plus près est un élément de contour et qu'il n'y a pas de redondance à cet endroit. Alors l'élément de la fonction de champ intégrée sera aussi un élément de contour de la surface résultante qui passera au même endroit que la surface de l'élément retenu.

2.2.2.3 Sélection des éléments pour le calcul

À partir de tous les éléments utiles de toutes les fonctions de champ individuelles pour un point donné dans l'espace, certains éléments sont sélectionnés pour le calcul de la valeur de la fonction de champ intégrée en ce point. Une première sélection est faite en conservant les éléments de non-contour qui ont la même orientation que l'élément minimum de non-contour. L'orientation des éléments est évaluée à l'aide du vecteur normal à la surface qui est conservé avec l'information de distance et de contour. Un élément de même orientation que l'élément minimum signifie que l'angle entre les deux vecteurs est inférieur à 90° . Cette condition est vérifiée à l'aide de l'inéquation 2.10.

$$\vec{N}_{\min_{NC}} \bullet \vec{N}_{i_{NC}} \geq 0 \quad (2.10)$$

À l'inéquation 2.10, le vecteur $N_{\min_{NC}}$ représente le vecteur normal de l'élément minimum de non-contour et le vecteur $N_{i_{NC}}$ représente les vecteurs normaux des autres éléments de non-contour. Un produit scalaire positif entre deux vecteurs indique qu'ils sont de même orientation.

Ensuite une recherche de l'élément minimum de non-contour ayant une orientation opposée (produit scalaire négatif) à l'élément minimum de non-contour est effectuée. Maintenant, à partir de la première sélection, tous les éléments qui ont une distance absolue plus grande que celle de l'élément minimum d'orientation opposée sont éliminés. Finalement, à partir des éléments restants, tous les éléments qui sont à une distance plus grande que la distance de redondance de l'élément minimum de non-contour sont éliminés.

À ce stade, les éléments restants forment l'ensemble solution des éléments qui serviront au calcul de la fonction de champ intégrée. Cette procédure de sélection a pour but d'éliminer du calcul les éléments qui ne correspondent pas à la même surface parce qu'ils sont d'orientation opposée ou bien parce qu'ils sont trop loin pour être considéré comme étant la même section de surface numérisée.

2.2.2.4 Calcul de la valeur intégrée

Le calcul des valeurs de la fonction de champ intégrée pour un élément donné est fait à partir des éléments restants dans l'ensemble solution élaboré à l'étape précédente. Le calcul de la distance à la surface est une moyenne de toutes les distances des éléments des autres fonctions de champ qui font partie de l'ensemble solution. Si, par exemple, il y a seulement deux éléments restants pour le calcul, alors la surface résultante intégrée passera à mi-chemin entre les deux surfaces originales du point de vue de l'élément où le calcul est effectué. Le vecteur normal à la surface n'est pas sauvegardé dans la

fonction de champ intégrée puisque cette information était utile au processus de fusion des surfaces seulement. La dernière étape de triangulation de la fonction de champ intégrée pour obtenir la surface polygonale explicite résultante n'utilise pas cette information. En ce qui concerne l'information de contour, cet élément est étiqueté comme un élément de non-contour, c'est-à-dire que le point le plus près de cet élément sur la surface intégrée ne fait pas partie du contour de cette surface.

Les étapes de l'intégration de plusieurs fonctions de champ en une seule se résument de la façon suivante :

1. Calculer les fonctions de champ de toutes les surfaces.
2. Calculer la boîte de contour et créer la grille de la fonction de champ intégrée.
3. Pour chaque élément de la fonction de champ intégrée, considérer tous les éléments utiles des autres fonctions pour ce point de l'espace.
4. Si aucun élément n'est utile, l'élément intégré n'est donc pas utile non plus à sa fonction. Fin de l'intégration pour cet élément.
5. Si un seul élément est utile, alors l'élément intégré est une copie conforme de cet élément utile. Fin de l'intégration pour cet élément.
6. Si plusieurs éléments sont utiles, trouver les éléments minimums de contour et de non-contour.
7. S'il n'y a pas d'élément de non-contour ou si la distance de l'élément minimum de contour est plus petite que celle de l'élément de non-contour moins la distance de redondance, alors l'élément intégré est une copie conforme de l'élément minimum de contour. Fin de l'intégration pour cet élément.
8. Créer un ensemble d'éléments avec tous les éléments de non-contour de même orientation que l'élément minimum de non-contour.
9. Trouver l'élément minimum de non-contour et d'orientation opposée à l'élément minimum de non-contour.
10. Éliminer de l'ensemble tous les éléments ayant une distance plus grande que celle de l'élément minimum de non-contour et d'orientation opposée.

11. Éliminer de l'ensemble tous les éléments qui ont une distance plus grande que celle de l'élément minimum de non-contour additionnée de la distance de redondance.
12. Calculer la distance moyenne des distances de tous les éléments restants dans l'ensemble. L'élément intégré est un élément de non-contour et sa distance est la moyenne calculée. Fin de l'intégration pour cet élément.

2.2.3 Implémentation informatique

L'implémentation informatique du processus de fusion des surfaces est réalisée à l'aide de la classe « SurfaceMergerImplicit ». Cette classe offre le support nécessaire pour calculer la fonction de champ intégrée à partir des fonctions de champ individuelles. Cette classe contient un paramètre important, il s'agit de la distance de redondance qui est utilisée à plusieurs reprises lors des différents calculs. Elle représente la distance maximale possible entre deux sections de surface pour que ces sections soient considérées comme redondantes. Deux surfaces à l'intérieur de la distance de redondance seront fusionnées en une seule alors que deux surfaces à l'extérieur de cette distance seront considérées comme deux surfaces distinctes dans le modèle final. Ce paramètre a un impact majeur sur l'intégration des surfaces implicites. Il doit être fixé en fonction de la précision globale du système de numérisation des surfaces.

Si la même surface est numérisée deux fois de suite, le résultat ne sera pas exactement le même à cause de l'imprécision du système de numérisation. Les deux surfaces ne seront pas exactement au même endroit dans l'espace. Il y a de fortes chances que les surfaces s'entrecroisent entre elles. La précision du système est quantifiable et elle provient de l'erreur des deux composants principaux du système global. La première erreur provient du système optique. Cette erreur est mesurée sur l'image d'un profil acquis et elle est de $\pm 0,25$ millimètre sur la profondeur mesurée. La seconde erreur provient du système de positionnement. Dans le cas du présent système de positionnement ultrasonique, l'imprécision comprend l'erreur linéaire sur X, Y, Z et l'erreur angulaire sur le tangage, le roulis et le lacet. L'erreur linéaire est mesurée à

l'aide d'un autre système de positionnement beaucoup plus précis et elle est de l'ordre de ± 1 millimètre sur la position dans l'espace. L'erreur angulaire est mesurée à l'aide d'étages de rotation. L'erreur du système optique étant plus faible, l'erreur du système global provient principalement du système de positionnement.

En pratique, la distance de redondance est fixée à 7 millimètres pour valider les algorithmes développés. Étant donné l'imprécision du système de positionnement, l'erreur entre les différents balayages peut atteindre cette amplitude selon la trajectoire de numérisation choisie. Les rotations du capteur sont les opérations qui produisent les erreurs les plus grandes entre les surfaces. Le système de positionnement est présentement dans une phase d'étalonnage afin d'améliorer la précision du capteur particulièrement en rotation pour éventuellement diminuer la valeur de la distance de redondance.

La figure 2.17 illustre l'intégration de surfaces implicites. Elle présente la fusion de deux demi-plans individuels en un seul plan intégré. La fusion de surfaces est illustrée à l'aide de surfaces de synthèse qui simulent une surface plane numérisée au moyen de deux balayages. Il y a donc deux surfaces au départ avec une zone de redondance au centre où les deux surfaces se superposent. Les figures 2.17A et 2.17B présentent ces surfaces sous des angles différents. Les deux surfaces sont parfaitement alignées selon le plan (X,Y) et elles sont légèrement décalées selon l'axe Z. La surface de gauche est légèrement en avant de celle de droite afin de simuler une erreur du système de numérisation et pour illustrer davantage le fonctionnement de l'intégration des surfaces. La valeur du décalage a été choisie plus petite que la distance de redondance afin que les deux surfaces soient fusionnées. Les figures 2.17A, 2.17C et 2.17E présentent les mêmes informations respectivement que les figures 2.17B, 2.17D et 2.17F mais sous des angles différents afin d'offrir une perspective différente pour mieux saisir le fonctionnement de l'intégration. La première série de figures présente les surfaces sous

un petit angle d'observation par rapport au vecteur normal aux plans alors que la seconde série les affiche avec un angle plus grand.

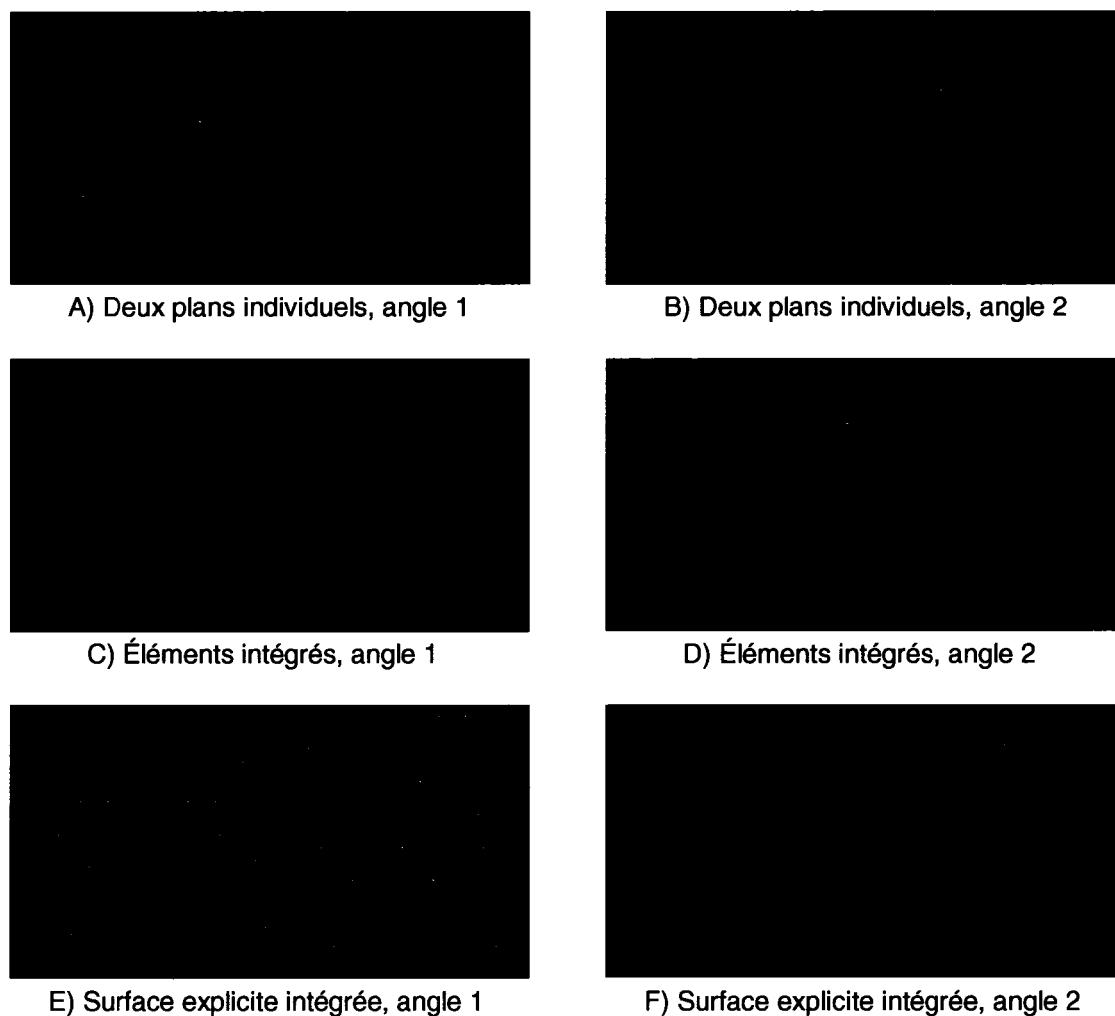


Figure 2.17 Intégration de deux surfaces

À partir des surfaces individuelles présentées aux figures 2.17A et 2.17B, les fonctions de champ de chacune des surfaces sont calculées individuellement. Ensuite les deux fonctions de champ sont intégrées en une seule. La fonction de champ globale de l'ensemble de la surface est représentée dans un format particulier aux figures 2.17C et 2.17D. La grille 3D de la fonction de champ n'est pas affichée sur les figures.

Seulement les éléments de la fonction de champ qui sont considérés sur la surface fusionnée sont affichés. Pour être considéré sur la surface, dans cet environnement discret, un élément doit avoir une distance absolue à la surface plus petite que la moitié de la résolution de la grille. Les éléments sont représentés par des sphères de différentes couleurs. Les sphères en gris représentent les éléments sur la surface qui ne font pas partie de son contour alors que les sphères en jaune représentent les éléments sur la surface qui en font partie. La surface implicite intégrée est au même endroit que les surfaces de départ dans les zones non redondantes, c'est-à-dire à gauche et à droite de la zone centrale. Au centre, dans la zone de redondance, l'emplacement de la surface intégrée correspond à la moyenne des deux surfaces individuelles. Les figures 2.17E et 2.17F présentent la surface polygonale explicite finale qui est le résultat de la triangulation de la fonction de champ globale intégrée. Ce résultat montre bien l'effet de moyenne dans la zone de redondance qui relie les deux extrémités non redondantes ensemble. L'algorithme de triangulation qui produit le maillage de la surface résultante à partir de la fonction de champ intégrée sera détaillé au chapitre suivant.

CHAPITRE 3

TRIANGULATION DU MODÈLE FINAL

À partir de la fonction de champ globale intégrée, un algorithme de triangulation est nécessaire pour obtenir une surface polygonale explicite. Ce chapitre présente deux méthodes différentes de triangulation de la fonction de champ qui ont été testées sur les données du MapScan. La première est une méthode surfacique étant donné qu'elle opère sur des surfaces et elle se nomme « marching triangle » [(Hilton & Illingworth, 1997), (Hilton & al, 1996a)]. La seconde nommée « marching cube » (Lorensen & Cline, 1987) est plutôt de type volumétrique parce qu'elle traite les points comme des volumes.

3.1 Triangulation « marching triangle »

La triangulation « marching triangle » est une méthode surfacique parce qu'elle débute à partir d'une portion de surface polygonale déjà existante et elle génère de nouveaux triangles qui s'y greffent jusqu'à l'atteinte des limites de la surface implicite. La surface polygonale devient de plus en plus grande au fur et à mesure que de nouveaux triangles sont ajoutés et se propagent dans toutes les directions, en suivant fidèlement la surface implicite, jusqu'à l'atteinte de la surface finale complète. Cette section présente l'algorithme du « marching triangle », son implémentation informatique ainsi que les améliorations apportées à l'algorithme de base pour tenter de l'adapter aux données du MapScan.

3.1.1 Algorithme du « marching triangle »

Au départ, une portion de surface explicite composée de triangles doit exister pour que l'algorithme du « marching triangle » puisse s'exécuter sur celle-ci afin de trianguler l'ensemble de la surface représentée par la fonction de champ intégrée. Cette surface

initiale peut contenir un seul ou plusieurs triangles et elle doit correspondre à une partie de la surface implicite. L'algorithme effectue d'abord une seule itération dans la liste des arêtes de la surface. Si par exemple le point de départ de l'algorithme est une surface qui contient un seul triangle alors la liste des arêtes en contient trois. Lorsqu'un second triangle est ajouté, la liste des arêtes est mise à jour et les arêtes non communes au triangle existant sont ajoutées à la fin de la liste pour qu'elles soient éventuellement traitées à leur tour. Dans cet exemple en particulier, la liste mise à jour contient cinq arêtes puisqu'une surface composée de deux triangles possède nécessairement une seule arête commune aux deux triangles. Pour chacune des arêtes de la surface, l'algorithme effectue plusieurs tests afin de déterminer si un nouveau triangle sera ajouté ou non à la surface. Si un triangle est ajouté à partir d'une arête en traitement, cette arête devient une des trois arêtes du nouveau triangle. La figure 3.1 présente le principe de l'algorithme du « marching triangle ».

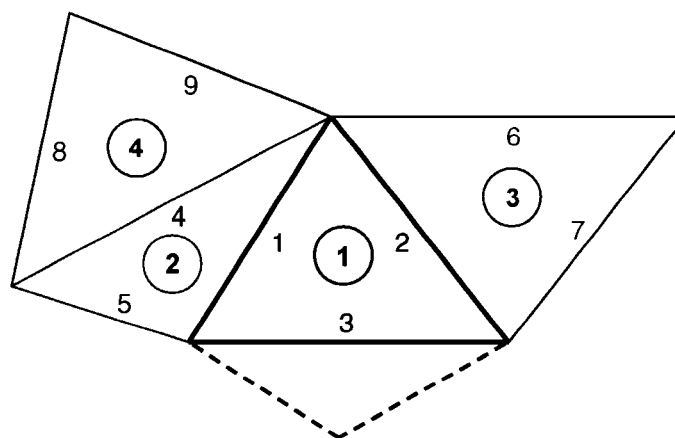


Figure 3.1 Principe du « marching triangle »

À la figure 3.1, la surface initiale, avant le début de l'algorithme, est composée du triangle 1 seulement. La liste des arêtes contient les arêtes 1 à 3. L'algorithme débute avec le traitement de l'arête 1 en premier et le triangle 2 est ajouté à la surface. Les arêtes 4 et 5 sont ajoutées à la liste des arêtes. Ensuite l'arête 2 est traitée et le triangle 3

est ajouté à la surface. Les arêtes 6 et 7 sont ajoutées à la liste des arêtes. Par la suite l'arête 3 est traitée et aucun triangle n'est ajouté à cette arête car il s'agit non seulement d'une arête de contour de la surface actuelle mais aussi d'une arête qui délimite le contour de la surface globale dans la direction du triangle en traits pointillés. Finalement l'arête 4 est traitée et le triangle 4 est ajouté à la surface. Les arêtes 8 et 9 sont à leur tour ajoutées à la liste des arêtes. L'exemple à la figure 3.1 se poursuivrait avec le traitement de l'arête 5 et ainsi de suite jusqu'au traitement de la dernière arête de la liste.

L'algorithme débute avec un premier test qui vérifie si l'arête en traitement est une arête de contour. Par définition, une arête de contour délimite le contour de la surface et elle appartient à un seul triangle comparativement aux autres arêtes qui sont communes à deux triangles. Si la surface possède un ou des trous, les arêtes autour d'un trou sont considérées comme des arêtes de contour au même titre que celles du contour extérieur de la surface puisqu'elles font partie d'un seul triangle. Les arêtes de contour sont distinguées des autres arêtes à la figure 2.3. À la figure 3.1, toutes les arêtes sont des arêtes de contour à l'exception des arêtes 1, 2 et 4. Si l'arête en traitement n'est pas une arête de contour, l'algorithme passe immédiatement à la prochaine arête dans la liste étant donné qu'il n'est pas possible d'ajouter un troisième triangle à une arête qui est déjà commune à deux triangles. Si l'arête délimite le contour, d'autres tests sont effectués sur celle-ci pour vérifier si un nouveau triangle doit être ajouté à cette arête.

À partir d'une arête de contour, la position d'un troisième sommet pour un nouveau triangle potentiel est évaluée. Une projection d'une distance constante est faite à partir du point milieu de l'arête vers l'extérieur de la surface et dans le plan du triangle de contour de la surface qui est composé de cette arête. Le calcul de la distance de projection est détaillé à la section 3.1.2. L'élément de la fonction de champ le plus près du point ainsi projeté et qui est sur la surface implicite est déterminé. Le nouveau triangle potentiel est composé de l'arête de contour dont les deux extrémités sont reliées

à ce nouveau point déterminé. D'autres tests seront effectués sur ce nouveau triangle potentiel afin de déterminer si effectivement ce triangle est ajouté à la surface. La figure 3.2 résume les étapes afin de déterminer le nouveau point formant un nouveau triangle potentiel à partir d'une arête et d'un triangle de contour d'une surface existante.

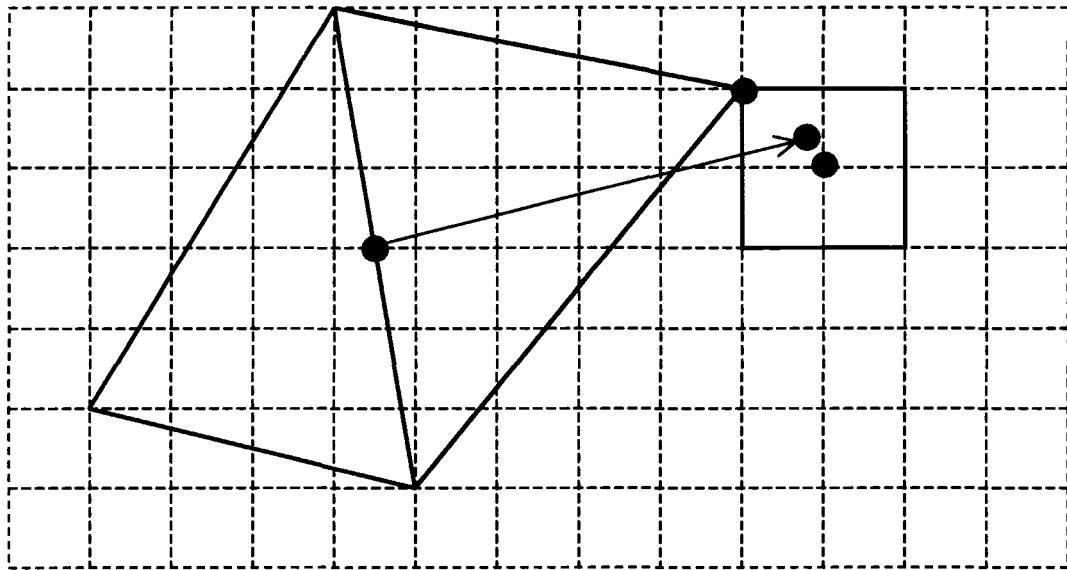


Figure 3.2 Détection d'un nouveau sommet possible

À la figure 3.2, la grille en trait pointillé représente la grille de la fonction de champ. L'exemple est présenté en 2D pour simplifier la compréhension mais le principe se généralise en 3D. Dans cet exemple, le triangle bleu est un triangle existant sur le contour de la surface. L'arête de contour en traitement est celle dont le point milieu est identifié par une sphère noire. La flèche illustre la projection vers l'extérieur et dans le plan du triangle bleu. Le point résultant de cette projection est indiqué par l'autre sphère noire. La sphère rouge indique l'élément de la fonction de champ le plus près du point projeté. Cet élément n'est cependant pas sur la surface implicite. Une recherche d'un élément sur la surface est effectuée en considérant les voisins immédiats de l'élément le plus près, soient ceux qui sont sur le carré rouge. L'élément identifié par la sphère bleue est sur la surface et la recherche se termine à ce point. Si aucun élément ne se trouve sur

la surface alors la recherche se poursuit en considérant les voisins immédiats de ceux qui sont sur le carré rouge. Le nouveau triangle potentiel est toujours formé de l'arête de contour d'où la projection est partie et des deux segments de droite en vert qui relient l'arête de contour au nouveau point.

Le test suivant vérifie l'information de contour au nouveau point. Si c'est un élément de contour, le contour de la surface est atteint dans cette direction et le nouveau triangle n'est pas ajouté à la surface. Dans ce cas l'algorithme se poursuit en traitant la prochaine arête dans la liste. Cette procédure qui consiste à ne pas ajouter de triangle à un élément de contour est discutable et elle pourrait être améliorée en y apportant une modification qui sera décrite un peu plus loin dans cette section. Si ce n'est pas un élément de contour alors les tests se poursuivent sur ce nouveau triangle potentiel. Le vecteur normal du nouveau triangle est calculé et il est comparé à celui du triangle de contour constitué de l'arête de contour qui est présentement en traitement pour vérifier si les deux triangles ont la même orientation. Deux triangles ont la même orientation si le produit scalaire entre les vecteurs normaux des triangles est positif. Si les triangles n'ont pas la même orientation alors le nouveau triangle n'est pas ajouté et l'algorithme se poursuit sur l'arête suivante dans la liste. Cela évite que la surface se replie sur elle-même ou encore qu'elle se torde d'un demi-tour à un endroit donné. Si les triangles ont la même orientation alors d'autres tests sont effectués sur le nouveau triangle potentiel pour vérifier s'il doit être ajouté.

Le prochain test consiste à définir une sphère qui passe par les trois sommets du nouveau triangle potentiel. Cette sphère circonscrite au triangle a pour centre le point de concours des médiatrices du triangle. La figure 3.3 présente deux exemples de cette sphère.

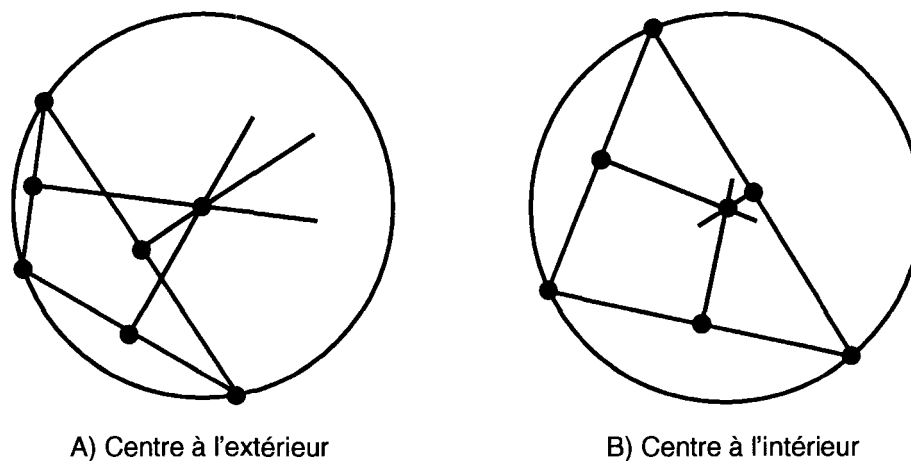


Figure 3.3 Sphère circonscrite à un triangle

Les cercles rouges à la figure 3.3 représentent des sphères en 3D pour le test. Pour un triangle donné, en bleu sur cette figure, les médiatrices sont tracées en vert. Celles-ci passent par le point milieu des arêtes et elles sont perpendiculaires à ces dernières. Les trois droites se croisent toujours en un seul point en rouge qui est aussi le centre de la sphère qui passe par les trois sommets du triangle. Le centre de la sphère peut être à l'extérieur ou à l'intérieur du triangle comme illustré aux figures 3.3A et 3.3B respectivement. Le rayon de la sphère est la distance entre le point de rencontre des médiatrices et un des sommets du triangle.

Une fois la sphère définie, le test vérifie s'il y a une partie de la surface existante qui est à l'intérieur de la sphère. Le triangle composé de l'arête qui est présentement en traitement est exclu de ce test puisqu'il est nécessairement à l'intérieur de la sphère. La configuration possible d'une surface qui décrit ce test est présentée à la figure 3.4A. S'il n'y a aucune partie d'aucun triangle qui est à l'intérieur de la sphère alors le nouveau triangle potentiel est ajouté à la surface. Ce test sert à vérifier s'il n'y a pas d'interférence entre le nouveau triangle et les autres triangles de la surface existante. Si une partie de la surface est à l'intérieur de la sphère alors le nouveau triangle n'est pas

ajouté. Avant de passer à l'arête suivante dans la liste, trois nouveaux triangles potentiels seront testés pour voir si l'un d'eux peut être ajouté à la surface.

Les trois nouveaux triangles possibles sont tous formés à partir de la même arête de contour en traitement et seulement le troisième sommet varie d'un triangle à l'autre. Le premier des trois cas est celui où le troisième sommet du triangle appartient à l'arête de contour qui précède immédiatement l'arête en traitement en suivant le contour de la surface. Le sommet recherché est évidemment celui qui n'appartient pas à l'arête en traitement puisque ces deux arêtes ont un sommet en commun. Ce cas est illustré à la figure 3.4B. Ce nouveau triangle subit le même test de la sphère circonscrite décrit précédemment. Cependant un second triangle est exclu du test et il s'agit du triangle défini par l'arête de contour précédente à celle en traitement. Si le test est réussi alors le nouveau triangle est ajouté à la surface et l'algorithme passe à l'arête suivante dans la liste.

Dans le cas contraire, le second des trois cas est évalué et il s'agit du cas où le troisième sommet du triangle appartient à l'arête de contour qui suit immédiatement l'arête en traitement en suivant le contour de la surface. Le sommet recherché est toujours celui qui n'appartient pas à l'arête en traitement puisque ces deux arêtes ont aussi un sommet en commun. Ce cas est illustré à la figure 3.4C. Ce nouveau triangle subit toujours le même test de la sphère circonscrite. Un second triangle est encore exclu du test et il s'agit du triangle défini par l'arête de contour suivante à celle en traitement. Si le test est réussi alors le nouveau triangle est ajouté à la surface et l'algorithme passe à l'arête suivante dans la liste. Sinon le troisième cas possible est évalué. Les tests sur les deux premiers cas décrits servent à vérifier s'il y a une possibilité de créer un nouveau triangle avec le triangle précédent ou le suivant en ajoutant seulement une arête car ce sont souvent ces triangles qui créent une interférence avec la sphère du nouveau triangle original testé.

Si l'algorithme se rend jusqu'à tester le troisième cas possible c'est qu'aucun triangle n'a encore été ajouté à la surface. Ce qui signifie que le test de la sphère circonscrite au triangle a échoué pour le nouveau triangle original ainsi que pour les deux cas précédents des nouveaux triangles possibles. Si le test de la sphère a échoué c'est qu'il y avait une partie d'au moins un triangle qui était à l'intérieur de la sphère dans chacun des cas. Le ou les triangles qui étaient à l'intérieur de la sphère dans le cas du nouveau triangle original seulement sont testés pour vérifier s'il y avait un ou des triangles de contour parmi eux ayant la même orientation que le triangle de contour composé de l'arête de contour en traitement. Les triangles 8 et 9 à la figure 3.15 (section 3.1.3.1 du document) sont des exemples des triangles recherchés par ce test. S'il n'y avait pas de triangle de contour avec la même orientation parmi eux alors aucun triangle ne sera ajouté à l'arête de contour en traitement et l'algorithme passe définitivement à l'arête suivante dans la liste. S'il y avait des triangles de contour avec la même orientation parmi eux alors pour chacun d'eux un nouveau triangle possible est testé.

Dans les deux sommets de l'arête de contour du triangle de contour qui interférait avec la sphère du nouveau triangle original, celui qui est le plus près de l'arête de contour en traitement est retenu comme troisième sommet du nouveau triangle de ce troisième cas possible. Ce cas est illustré à la figure 3.4D. Le même test de la sphère est encore une fois appliqué sur ce nouveau triangle potentiel. Dans ce cas un second triangle est exclu du test et cette fois-ci il s'agit du triangle composé du troisième sommet du nouveau triangle. Si le test est réussi pour un des triangles considérés alors le nouveau triangle est ajouté à la surface et l'algorithme passe à l'arête suivante dans la liste. Dans le cas contraire, si le test échoue pour tous les triangles considérés à cette étape, aucun triangle ne sera ajouté à l'arête de contour en traitement et l'algorithme passe définitivement à l'arête suivante dans la liste. Le troisième cas décrit sert à vérifier s'il y a une possibilité de relier par un pont de triangles deux parties non voisines du contour de la surface qui ont la même orientation. À l'occasion, ce sont des triangles de contour non voisins et de même orientation qui interfèrent avec la sphère du nouveau triangle original. Si un

nouveau triangle est ajouté à cette étape alors un trou dans la surface est nécessairement créé. Ce trou sera comblé s'il y a lieu lorsque les arêtes de contour du trou seront traitées à leur tour. La figure 3.4 illustre les différents cas possibles pour ajouter un nouveau triangle à la surface.

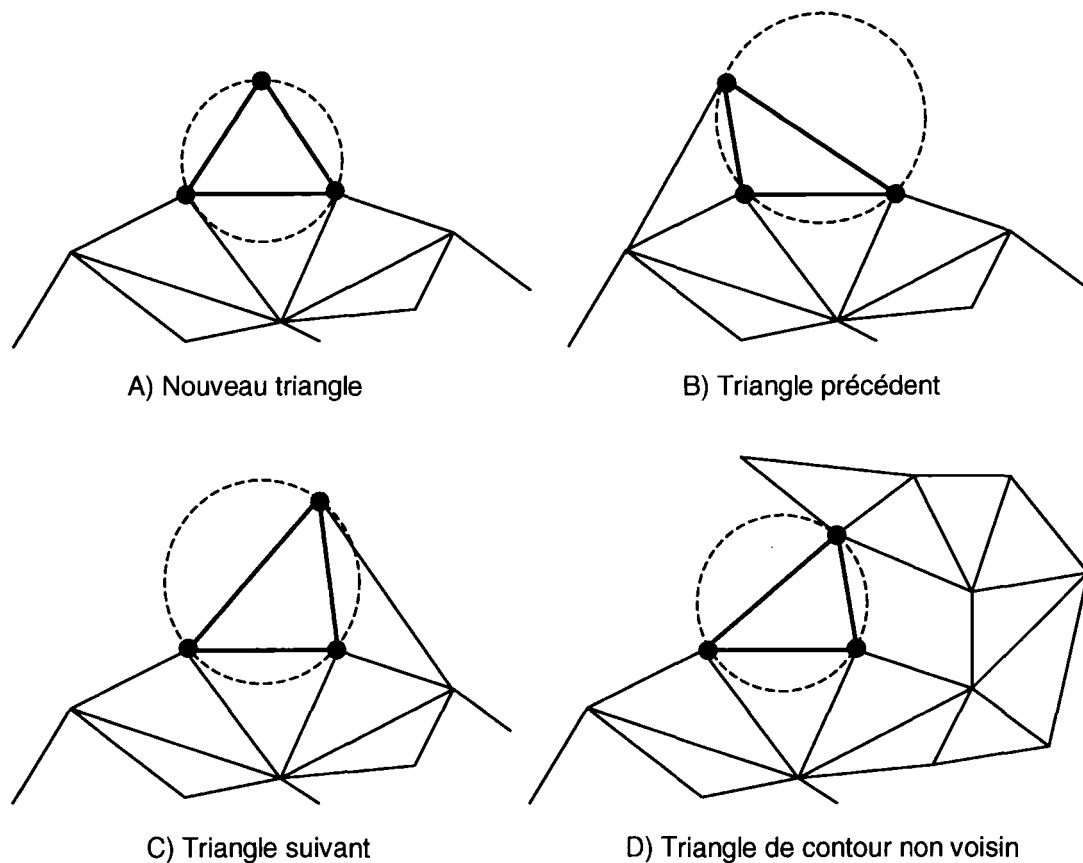


Figure 3.4 Nouveaux triangles potentiels à ajouter à la surface

La figure 3.4A présente le cas du nouveau triangle original dont le nouveau sommet est issu de la recherche de l'élément sur la surface implicite qui est le plus près de la projection effectuée. Les figures 3.4B et 3.4C présentent respectivement les cas des nouveaux triangles créés à l'aide des triangles de contour précédent et suivant de l'arête de contour en traitement. La figure 3.4D présente le cas du nouveau triangle créé à l'aide du sommet de contour le plus près de l'arête de contour en traitement qui provient

d'un triangle de contour et de même orientation qui interférerait avec la sphère du nouveau triangle original. Dans les exemples à la figure 3.4, les arêtes et les sommets en bleu représentent l'arête de contour en traitement et ses sommets. Les arêtes et les sommets en rouge représentent des nouveaux éléments à ajouter à la surface lorsque le nouveau triangle est ajouté. Les arêtes et les sommets en vert représentent des éléments déjà existants de la surface mais ces éléments constitueront aussi le nouveau triangle ajouté. Les cercles en traits pointillés représentent la sphère circonscrite pour les tests. Dans les quatre cas présentés à la figure 3.4, le test de la sphère serait réussi et les nouveaux triangles seraient ajoutés à la surface. La figure 3.5 résume les étapes de l'algorithme du « marching triangle » à l'aide d'un ordinogramme.

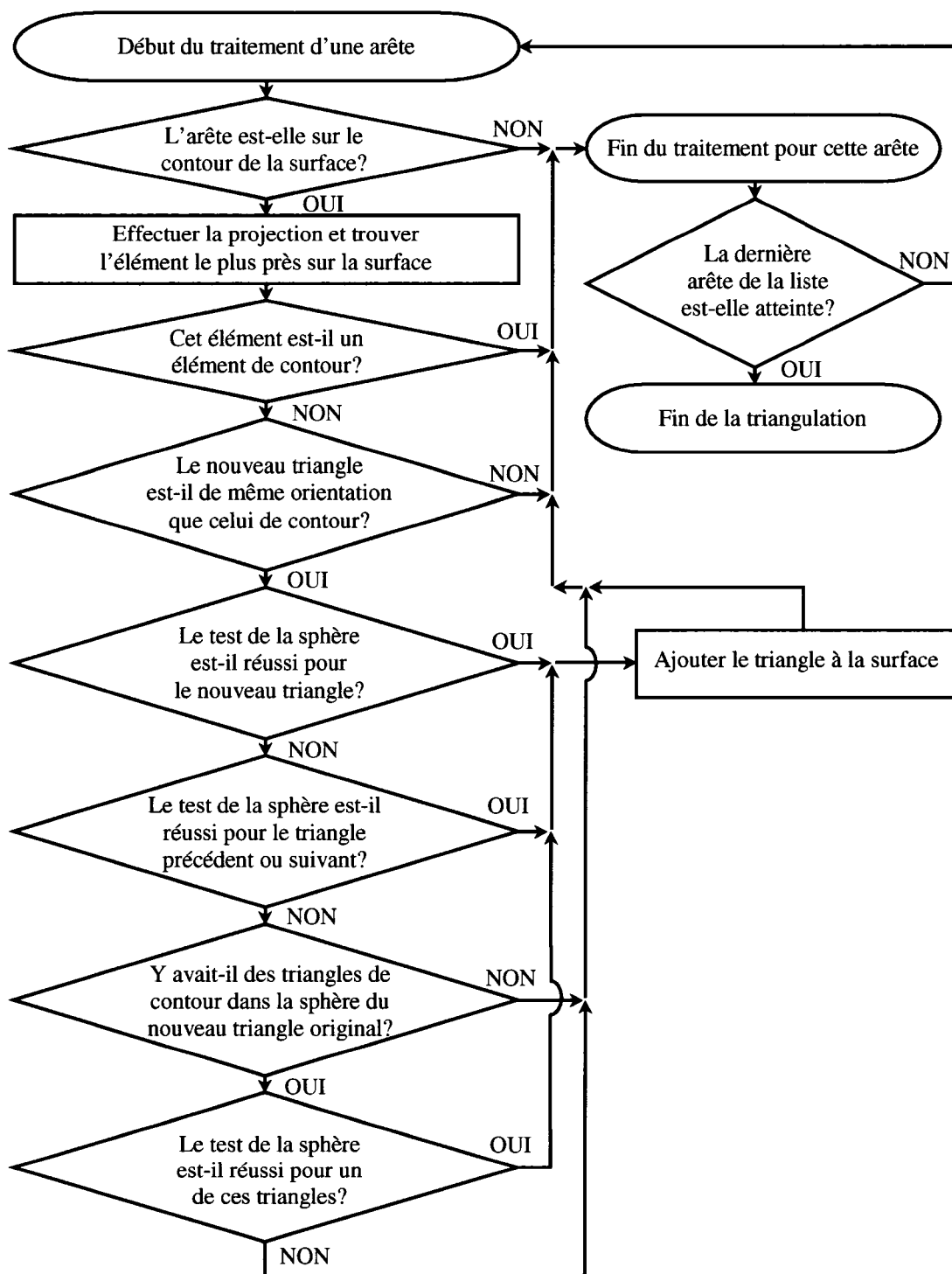
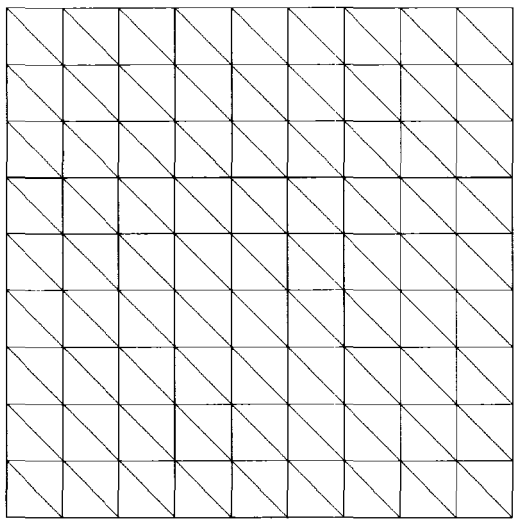
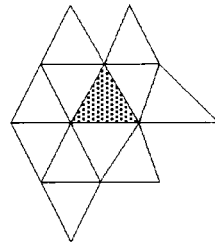


Figure 3.5 Étapes de l'algorithme du « marching triangle »

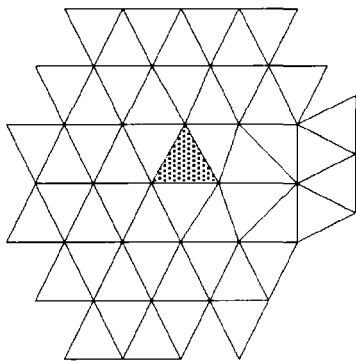
La figure 3.6 illustre la progression de l'algorithme du « marching triangle » sur une surface de synthèse qui représente un plan.



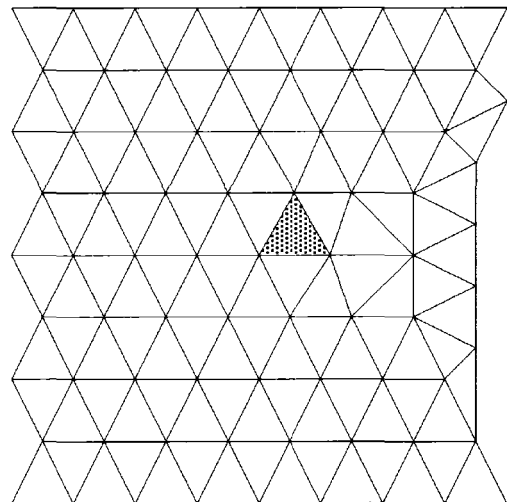
A) Surface initiale



B) Surface triangulée avec 12 triangles



C) Surface triangulée avec 51 triangles



D) Surface triangulée avec 118 triangles

Figure 3.6 Progression du « marching triangle » sur un plan

La figure 3.6A présente la surface de synthèse initiale. La fonction de champ de cette surface a été calculée et ensuite l'algorithme du « marching triangle » a été lancé sur

cette fonction de champ. Les figures 3.6B, 3.6C et 3.6D présentent la progression de l'algorithme sur la surface résultante avec respectivement 12, 51 et 118 triangles. La figure 3.6D présente la surface résultante finale suite à l'arrêt de l'algorithme. Le triangle ombragé aux figures 3.6B, 3.6C et 3.6D indique le premier triangle qui a été ajouté manuellement à la surface avant de lancer l'algorithme de triangulation. La surface résultante finale est relativement régulière et elle est composée de triangles semblables qui sont pratiquement tous équilatéraux. Ce phénomène dépend essentiellement du triangle de départ ainsi que de la distance de projection choisie. Le choix de ces paramètres sera discuté à la section suivante.

Il semble manquer une colonne de triangles de chaque côté de la surface finale à la figure 3.6D pour compléter le plan original à la figure 3.6A. Ce phénomène est causé par le critère qui spécifie que si l'élément choisi pour être le troisième sommet du nouveau triangle est un élément de contour de la fonction de champ alors le nouveau triangle n'est pas ajouté. Pour la même raison, si la surface contient des trous ils seront légèrement agrandis car une rangée de triangles sera manquante sur le contour des trous. À grande échelle, sur un modèle qui contient plusieurs milliers de triangles, le fait qu'il manque une rangée de triangles sur le contour de la surface et des trous est très négligeable visuellement. Ce test pourrait toutefois être modifié afin de corriger la situation. Même si l'élément est un élément de contour, il est tout de même considéré comme étant sur la surface. Le triangle devrait être ajouté et les nouvelles arêtes de contour de ce triangle devraient être étiquetées afin qu'elles ne soient pas considérées pour l'ajout de nouveaux triangles supplémentaires. De cette façon le critère d'arrêt de la triangulation dans cette direction serait respecté et il ne manquerait pas une rangée de triangles sur le contour de la surface.

3.1.2 Implémentation informatique

Afin d'implémenter l'algorithme du « marching triangle », les valeurs de plusieurs paramètres ont été sélectionnées pour permettre le bon fonctionnement de l'algorithme.

Comme celui-ci doit débiter avec une portion de surface explicite, dans tous les cas cette surface est composée d'un seul triangle. Ce triangle provient d'une des surfaces initiales et il s'agit du triangle le plus équilatéral possible rencontré dans toutes les sections non redondantes des surfaces initiales. Un triangle est considéré comme étant dans une zone redondante si au moins un autre triangle d'une autre surface initiale est à une distance plus petite que la distance de redondance établie de ce triangle. Pour vérifier ce critère, les distances sont calculées entre les sommets des triangles. Pour déterminer le triangle le plus équilatéral, trois différences sont calculées et elles sont transformées en valeurs absolues pour chacun des triangles. Il s'agit des différences entre les longueurs de deux des côtés du triangle. Ensuite la somme de ces trois différences absolues est calculée et en choisissant le triangle qui minimise cette somme, le triangle le plus équilatéral est obtenu. La figure 3.7 présente ce principe.

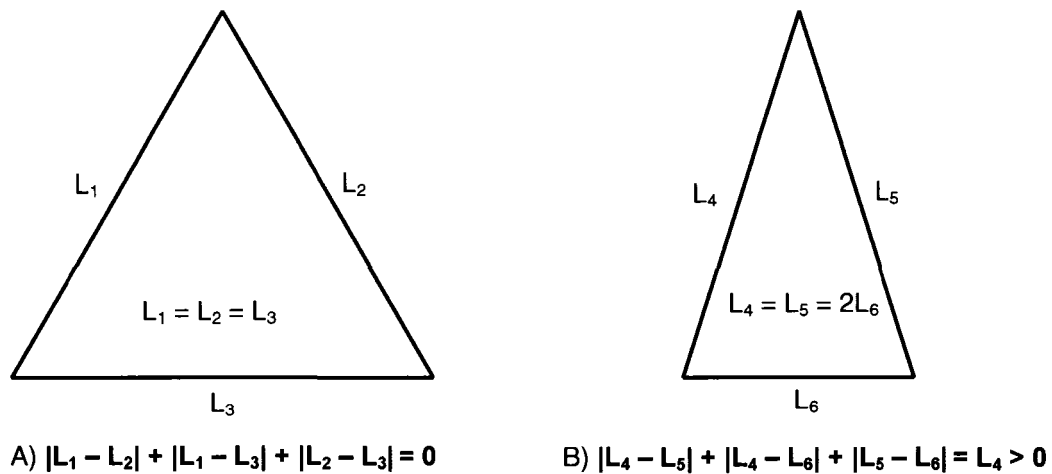


Figure 3.7 Caractéristique d'un triangle équilatéral

Le périmètre du triangle sélectionné est évalué et il est comparé au périmètre moyen des triangles de l'ensemble des surfaces initiales. Si la différence entre le périmètre moyen et celui de ce triangle est inférieure à 20% alors le triangle devient le premier triangle de la surface résultante. Sinon, le triangle est rejeté et le périmètre du second triangle le plus équilatéral est comparé au périmètre moyen et ainsi de suite jusqu'à l'obtention

d'un triangle adéquat. L'inéquation 3.1 est utilisée pour calculer la différence entre les périmètres et déterminer si le triangle est adéquat.

$$\frac{|P_i - P_M|}{P_M} \leq 0,2 \quad (3.1)$$

À l'inéquation 3.1, P_M représente le périmètre moyen de tous les triangles et P_i celui du triangle à vérifier. La sélection d'un premier triangle pratiquement équilatéral pour la surface résultante est un des facteurs qui améliorent la qualité du résultat final de l'algorithme. La surface est alors composée de triangles pratiquement identiques en termes de forme et de grandeur et ce particulièrement dans les zones ayant une faible courbure. La restriction sur le périmètre du premier triangle choisi permet d'obtenir des triangles de grandeur moyenne similaire sur la surface résultante comparativement aux surfaces initiales.

Un second facteur important qui influence la qualité du résultat final est la distance de projection choisie pour évaluer le troisième sommet du nouveau triangle à ajouter à la surface à partir de l'arête de contour en traitement. Cette distance est choisie de façon à obtenir une surface résultante la plus uniforme possible composée de triangles pratiquement équilatéraux et de mêmes dimensions que le triangle de départ. Si ce premier triangle était parfaitement équilatéral, la longueur des ses côtés serait égale au tiers de son périmètre. Si le troisième sommet d'un nouveau triangle était exactement le point projeté alors la distance de projection correspondrait à la hauteur du triangle. Pour un triangle équilatéral, la hauteur du triangle représente une fraction constante de la longueur de ses côtés et cette fraction est $\sqrt{3}/2$. La figure 3.8 présente cette relation.

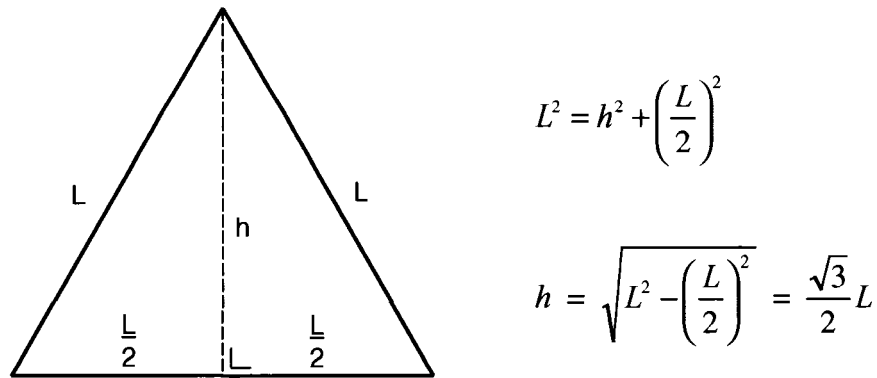


Figure 3.8 Relation entre la hauteur et les côtés d'un triangle équilatéral

La distance de projection choisie est donc le tiers du périmètre du triangle de départ multiplié par $\sqrt{3}/2$. Pour le bon fonctionnement de l'algorithme et afin que la surface puisse être triangulée, la distance de projection doit être supérieure à une distance minimale qui est le double de la résolution de la grille. Si la distance de projection est inférieure à ce seuil, le nouveau sommet recherché sera pratiquement toujours un des deux sommets de l'arête de contour en traitement. Le nouveau triangle serait donc incohérent et aucun triangle ne serait ajouté à la surface. La distance de projection calculée est comparée à ce seuil et si elle est inférieure alors elle est mise égale au seuil de comparaison. Si cette situation se présente, la surface résultante sera moins uniforme car la distance entre le point projeté et le point le plus près sur la surface, déterminé lors de la recherche, sera plus grande que si la distance entre les éléments de la grille était beaucoup plus petite par rapport à la distance de projection. Cette méthode pour calculer la distance de projection est efficace et elle donne de bons résultats.

D'autres méthodes pour calculer la distance de projection ont été testées et une seconde méthode qui est aussi efficace que la première a été retenue. Selon la définition de cet algorithme, la distance de projection doit être constante mais la seconde méthode résulte en une distance variable calculée dynamiquement pour chaque arête traitée. La distance de projection est égale à la longueur de l'arête multipliée par la même fraction que

précédemment. Ceci anticipe que le nouveau triangle sera équilatéral si, encore une fois, le point projeté était exactement le troisième sommet du nouveau triangle. Les deux méthodes donnent des résultats semblables et pour bien les comparer et vérifier les différences entre elles, il faut effectuer un test avec une surface relativement plane et une grille très précise pour la fonction de champ. Dans ce cas le point projeté sera déplacé légèrement et la distorsion sur les triangles engendrée par la résolution de la grille sera minimisée. La première méthode décrite avec une distance de projection constante a tendance à générer des triangles un peu moins équilatéraux que la seconde méthode avec une distance de projection variable. Par contre les triangles de la première méthode ont une plus faible distribution de leur périmètre autour de la valeur moyenne des périmètres des triangles de la surface résultante que ceux de la seconde méthode. Tout dépendant du critère prioritaire entre la forme équilatérale et la grandeur uniforme des triangles, l'une ou l'autre des méthodes décrites peut être utilisée.

Pour réaliser la projection et ainsi déterminer le point projeté, un vecteur est créé avec l'arête de contour en traitement. Il s'agit du vecteur **BC** à la figure 3.9. Le sens de ce vecteur est important et il doit tenir compte de l'ordre de définition des sommets du triangle pour obtenir une projection vers l'extérieur du triangle plutôt que vers l'intérieur. Le produit vectoriel du vecteur **BC** créé avec l'arête par le vecteur **N** qui est normal au triangle est calculé et normalisé. Le vecteur unitaire résultant, qui est illustré par la petite flèche verte en caractère gras à la figure 3.9, est multiplié par la distance de projection pour donner le vecteur **PP'**. Ensuite ce vecteur est ajouté au point milieu de l'arête pour obtenir le point projeté **P'**.

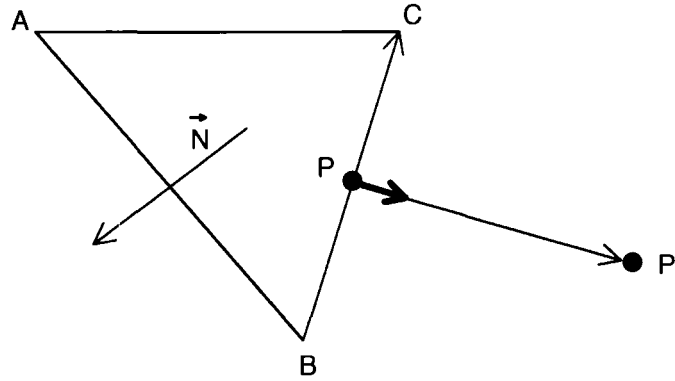


Figure 3.9 Projection d'un nouveau point

L'équation 3.2 est utilisée pour obtenir les coordonnées du point projeté P' . Les variables utilisées réfèrent à celles de la figure 3.9. La distance d représente la distance de projection.

$$\vec{P}' = \vec{P} + d \left(\frac{\vec{BC} \times \vec{N}}{|\vec{BC} \times \vec{N}|} \right) \quad (3.2)$$

Le point projeté dans l'espace 3D se trouve à l'intérieur d'un cube imaginaire formé par huit éléments voisins de la fonction de champ. Ces huit éléments sont déterminés en arrondissant les coordonnées du point projeté aux multiples inférieur et supérieur de la résolution de la grille de la fonction de champ puisque la grille est alignée sur ces multiples. L'élément le plus près du point projeté est sélectionné parmi les huit possibles en prenant celui qui est à la distance minimale du point. Les coordonnées de cet élément sont vérifiées pour s'assurer qu'il est à l'intérieur du domaine de la fonction de champ. Dans le cas où l'élément serait à l'extérieur du domaine, l'élément le plus près du point projeté devient l'élément à la limite inférieure ou supérieure du domaine, selon le cas, dans les axes concernés. Si cet élément se situe sur la surface implicite alors il s'agit du troisième sommet du nouveau triangle. Pour être considéré sur la

surface, un élément doit avoir une distance absolue à la surface inférieure ou égale à la moitié de la résolution de la grille.

Si l'élément le plus près n'est pas sur la surface alors une recherche est lancée parmi les voisins immédiats de cet élément pour trouver un élément sur la surface. Si un ou des éléments sur la surface sont trouvés parmi les voisins immédiats, l'élément le plus près du point projeté est conservé et il devient le troisième sommet du nouveau triangle. Sinon, la recherche se poursuit avec les voisins immédiats des premiers voisins déjà sélectionnés et ainsi de suite, toujours en s'éloignant de l'élément initial dans l'espace 3D, jusqu'au moment où un élément sur la surface soit trouvé. Il se pourrait que le troisième sommet ainsi déterminé du nouveau triangle soit en réalité le même point qu'un des deux autres sommets du nouveau triangle. Une comparaison est faite entre les coordonnées des points pour vérifier cette éventualité. Si c'est le cas alors le point est rejeté et la recherche se poursuit.

Pour être en mesure d'effectuer le test de la sphère, il faut définir la sphère circonscrite au triangle avec les coordonnées de son centre ainsi que la longueur de son rayon. Comme les médiatrices du triangle se rencontrent toutes en un seul point au centre de la sphère, il suffit de calculer le point de rencontre entre deux médiatrices pour obtenir les coordonnées du centre de la sphère. La figure 3.10 présente les paramètres utiles au calcul du centre de la sphère.

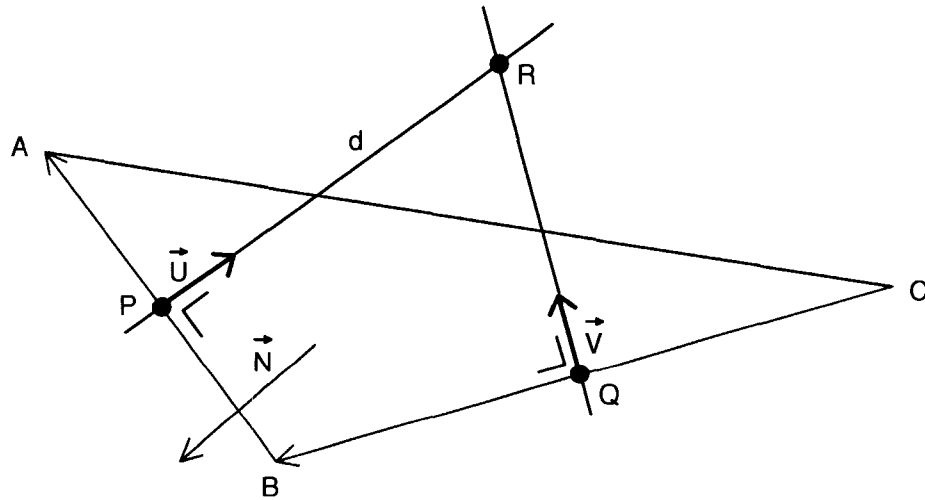


Figure 3.10 Paramètres utiles au calcul du centre de la sphère

À la figure 3.10, le vecteur normal N au nouveau triangle ABC est calculé selon l'équation 3.3.

$$\vec{N} = \overrightarrow{AB} \times \overrightarrow{AC} \quad (3.3)$$

Les points milieux P et Q des arêtes AB et BC sont calculés selon la moyenne des coordonnées des sommets des arêtes. À titre d'exemple pour P et Q , le point P est calculé selon l'équation 3.4.

$$P = \frac{A+B}{2} \quad (3.4)$$

Les deux arêtes dont les points milieux ont été calculés sont vues comme les vecteurs BA et CB et les produits vectoriels entre chacune des arêtes et le vecteur normal N au triangle sont calculés et normalisés pour obtenir les vecteurs U et V . À titre d'exemple pour U et V , Le vecteur U est calculé selon l'équation 3.5.

$$\vec{U} = \frac{\overrightarrow{BA} \times \vec{N}}{|\overrightarrow{BA} \times \vec{N}|} \quad (3.5)$$

Deux vecteurs unitaires sont ainsi obtenus et lorsque ces vecteurs sont positionnés avec leur point de départ sur les points milieux P et Q respectifs des deux arêtes, ils pointent tous les deux dans la direction du centre R de la sphère. Les deux équations paramétriques des droites qui passent par les points milieux des arêtes et qui se rencontrent au centre de la sphère sont déterminées avec les coordonnées des points milieux P et Q et celles des vecteurs unitaires U et V calculés. À titre d'exemple pour les deux droites, l'équation 3.6 est l'équation de la droite qui passe par le point P dans la direction du vecteur U .

$$\vec{X} = \vec{P} + t\vec{U} \quad t \in \mathfrak{R} \quad (3.6)$$

Le vecteur X à l'équation 3.6 représente tous les points de la droite et le paramètre t est un nombre réel. Un système à deux équations et deux inconnus est déterminé en utilisant les équations paramétriques des deux droites pour trouver leur point de rencontre au centre R de la sphère. Ce système est décrit par les équations 3.7 et 3.8.

$$P_x + tU_x = Q_x + sV_x \quad s, t \in \mathfrak{R} \quad (3.7)$$

$$P_y + tU_y = Q_y + sV_y \quad s, t \in \mathfrak{R} \quad (3.8)$$

Il existe deux nombres réels pour les inconnus t et s aux équations 3.7 et 3.8 qui satisfont ces équations. Ces nombres représentent les grandeurs des vecteurs \overrightarrow{PR} et \overrightarrow{QR} à la figure 3.10 et ils sont la solution pour déterminer le centre R de la sphère. Un seul des deux paramètres est nécessaire pour déterminer le centre R . Le paramètre s est isolé dans l'équation 3.7 et ensuite il est remplacé dans l'équation 3.8 par sa valeur. Le

paramètre t est finalement isolé et exprimé en fonction des variables connues dans l'équation 3.8. L'équation 3.9 présente le résultat du système pour le paramètre t .

$$t = d = \frac{(Q_y - P_y)V_x + (P_x - Q_x)V_y}{U_yV_x - U_xV_y} \quad (3.9)$$

Le paramètre t à l'équation 3.9 est aussi illustré comme étant la distance d entre les points P et R à la figure 3.10. Les coordonnées du centre R sont évaluées à l'aide de l'équation 3.6 en remplaçant le paramètre t par la valeur obtenue de l'équation 3.9. Le rayon de la sphère est simplement la distance entre le centre R et un des sommets du triangle.

Chacun des triangles de la surface est testé pour vérifier si une partie du triangle est à l'intérieur de la sphère. Une méthode numérique approximative plus simple et plus rapide a été préférée à une méthode analytique exacte pour réaliser ce test. La contrainte de surface sur le nouveau triangle est légèrement moins sévère avec la méthode numérique puisqu'il pourrait exister une infime partie d'un triangle dans la sphère qui ne serait pas détectée par la méthode. La boîte de contour de la sphère est calculée et un premier test rapide vérifie s'il y a une intersection entre celle-ci et celles des triangles individuellement. Ce test permet d'éliminer presque tous les triangles puisque seulement quelques triangles sont susceptibles d'interférer avec la sphère. Un second test rapide est effectué sur les triangles isolés du test précédent. La distance entre chacun des sommets du triangle et le centre de la sphère est évaluée. Si l'une d'elles est inférieure au rayon de la sphère alors le triangle interfère avec celle-ci.

La méthode numérique approximative est appliquée seulement sur les triangles dont l'interférence avec la sphère est encore inconnue suite aux deux tests rapides précédents. L'équation paramétrique du plan du triangle est déterminée avec un des

sommets du triangle et deux vecteurs formés avec les deux arêtes qui partent de ce sommet. La figure 3.11 illustre ce principe.

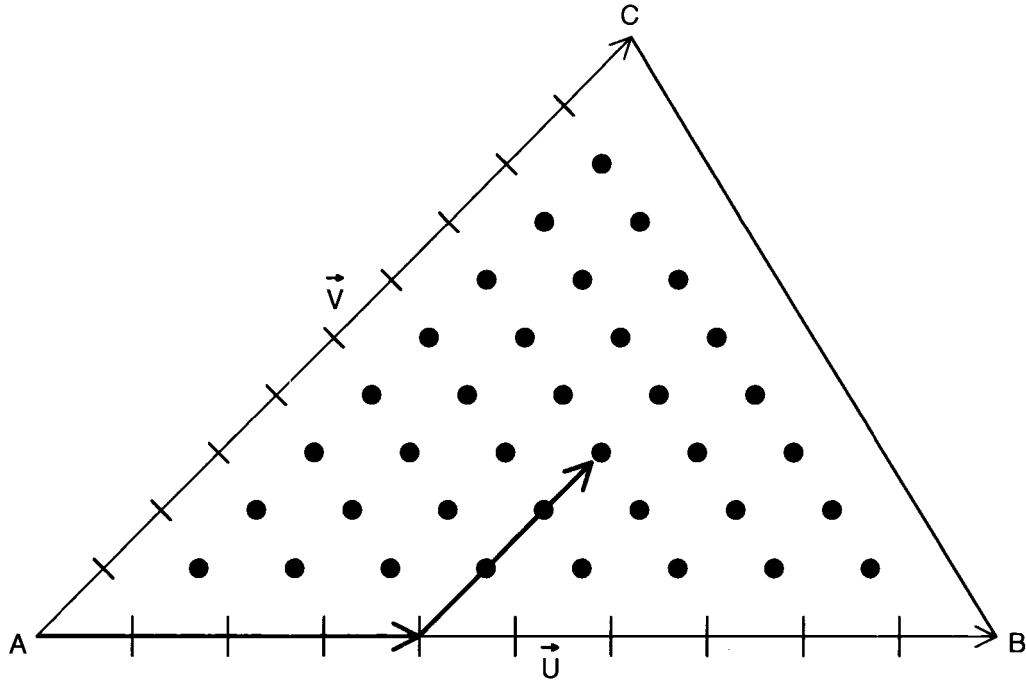


Figure 3.11 Discretisation du plan d'un triangle

L'équation du plan du triangle ABC à la figure 3.11 en fonction de son sommet A et des vecteurs U et V s'écrit selon l'équation 3.10.

$$\vec{X} = \vec{A} + t\vec{U} + s\vec{V} \quad t \in \mathcal{R}(0..1); s \in \mathcal{R}(0..1-t) \quad (3.10)$$

À l'équation 3.10, selon les nombres réels t et s , le vecteur X représente tous les points possibles sur le plan du triangle. Si t et s sont limités au domaine spécifié à l'équation 3.10, alors cette équation est bornée dans les deux axes engendrés par les vecteurs U et V pour représenter seulement la région du plan décrit par le triangle ABC . Les coordonnées de plusieurs points discrets à l'intérieur du triangle sont évaluées à l'aide de l'équation paramétrique où les variables t et s sont limitées à des sauts discrets de

0,1. Par exemple, le point obtenu à la figure 3.11 selon le vecteur bleu qui représente une fraction du vecteur U et le vecteur vert qui représente une fraction du vecteur V correspond à $(t = 0,4 ; s = 0,3)$. Le nombre de points évalués par triangle est ainsi fixé à 66 incluant les trois sommets qui ont déjà été évalués au test précédent. Ce nombre provient du domaine et de la discrétisation des variables t et s . Par exemple, si $t = 0$ alors 11 points seront évalués pour $s = 0..1$ par saut de 0,1. Si $t = 0,1$ alors 10 points seront évalués pour $s = 0..0,9$ et ainsi de suite jusqu'à $t = 1$ où 1 seul point sera évalué pour $s = 0..0$. Le nombre 66 correspond à la somme des nombres 1 à 11 inclusivement. La distance entre les points est constante à l'intérieur d'un même triangle mais elle est variable d'un triangle au suivant selon la longueur de ses arêtes. Pour chacun des points évalués à l'intérieur d'un triangle, la distance entre le point et le centre de la sphère est calculée et si elle est inférieure au rayon de la sphère alors une partie de ce triangle est nécessairement à l'intérieur de la sphère.

Une autre approche aurait pu être adoptée pour discrétiser les triangles. La distance entre les points évalués aurait pu être fixée à une valeur constante pour tous les triangles mais le nombre de points aurait été variable d'un triangle à l'autre. Au sujet de la série de points évalués, l'approche choisie assure une constance dans le temps d'exécution mais l'erreur induite par la méthode numérique est variable d'un triangle à l'autre. À l'inverse, cette autre approche suggérée assure une erreur induite constante mais le temps d'exécution est variable. Le choix de la première approche a été fait en fonction de l'importance des impacts des différentes approches sur les résultats obtenus. Le temps de calcul de cet algorithme est relativement long. Dans ce contexte, une évaluation efficace du temps de calcul a été préférée au contrôle d'une erreur qui ne varie pas énormément étant donné que la grandeur des triangles générés est relativement constante sur la surface résultante.

Pour vérifier si une arête est sur le contour d'une surface, il suffit de vérifier si cette arête appartient à un seul ou à deux triangles. Si l'arête appartient à un seul triangle

alors elle est sur le contour de la surface et si elle appartient à deux triangles alors elle n'est pas sur le contour. Le processus pour trouver les arêtes de contour précédentes et suivantes à celle en traitement, en suivant le contour de la surface afin de former les nouveaux triangles présentés aux figures 3.4B et 3.4C, est identique dans les deux cas. La seule différence est le sommet de départ de l'arête en traitement pour la recherche. Par convention, le résultat de la recherche à partir du premier sommet de l'arête en traitement indique les arêtes précédentes et celui à partir du second sommet indique les arêtes suivantes. Il est possible qu'une arête de contour possède plusieurs arêtes de contour précédentes ou suivantes si la surface contient des triangles reliés par un seul sommet. La figure 3.12 présente différentes configurations possibles pour les arêtes de contour voisines.

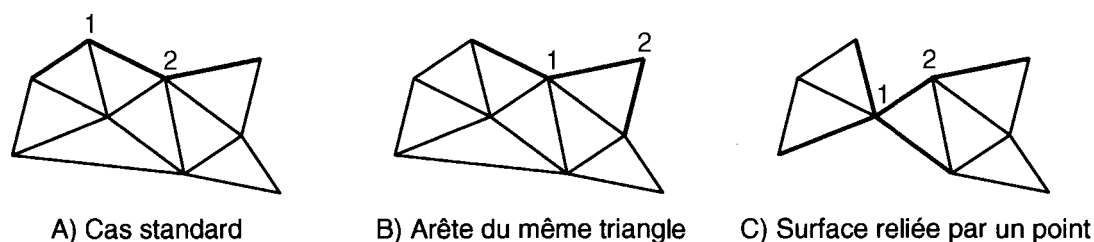


Figure 3.12 Configurations des arêtes de contour voisines

À la figure 3.12, l'arête en traitement ainsi que les numéros de ses sommets sont dessinés en rouge, les arêtes précédentes sont en bleu et les suivantes en vert. La figure 3.12A présente le cas standard où il existe une seule arête précédente et suivante qui appartiennent aux triangles précédent et suivant. La figure 3.12B présente le cas où l'arête suivante appartient au même triangle que l'arête en traitement. La figure 3.12C présente le cas où la surface est reliée par un seul point à un endroit donné. Cet exemple contient trois arêtes de contour voisines comme candidates à l'arête précédente.

À partir d'un sommet de l'arête de contour en traitement, une recherche est faite pour trouver toutes les arêtes de contour reliées à ce sommet. Ensuite certaines arêtes sont

exclues du résultat de la recherche. Il s'agit de l'arête présentement en traitement et, selon le cas, possiblement d'une arête qui est commune au même triangle que celle en traitement. À la figure 3.12B, l'arête suivante serait exclue et aucun triangle suivant ne serait généré puisque ce triangle existe déjà. À la figure 3.12C, une des trois arêtes candidates à l'arête précédente serait aussi exclue pour la même raison.

Un sommet d'une arête de contour a nécessairement un nombre pair d'arêtes de contour qui y sont reliées. Si les arêtes sont au nombre de deux (figure 3.12A et 3.12B), ce qui représente le cas le plus fréquent, alors une arête est exclue (l'arête en traitement) et il existe seulement une arête précédente ou suivante. Si les arêtes sont au nombre de quatre (figure 3.12C) ou plus alors deux arêtes sont exclues et le nombre restant d'arêtes est le nombre d'arêtes candidates pour être l'arête précédente ou suivante. Un sommet qui possède quatre arêtes de contour est un cas rencontré occasionnellement.

Dans tous les cas, le sommet non commun à l'arête en traitement de l'arête précédente ou suivante est sélectionné comme candidat au troisième sommet du nouveau triangle et il doit subir le test de la sphère. Si le sommet possède plus de deux arêtes de contour alors plusieurs nouveaux triangles seront évalués. Un test de continuité supplémentaire sera effectué sur les triangles avant de les ajouter à la surface s'ils ont réussi le test de la sphère. Ce test assure une continuité de la surface entre le nouveau triangle et le triangle de contour de l'arête précédente ou suivante. Il permet entre autre d'éliminer une des deux arêtes restantes comme candidates à l'arête précédente à la figure 3.12C. La figure 3.13 présente le principe de continuité de la surface entre les triangles.

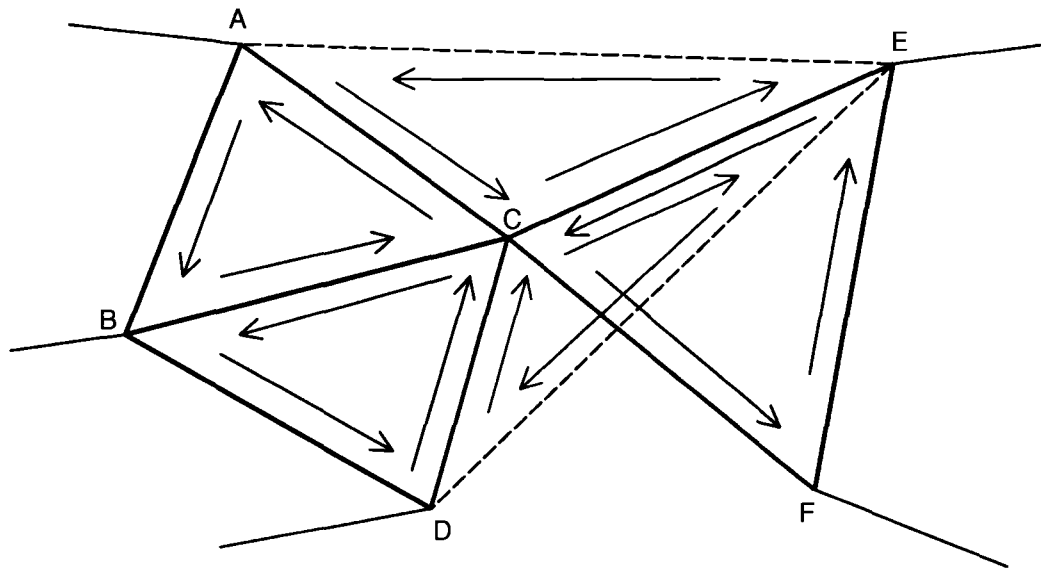


Figure 3.13 Continuité de la surface entre les triangles

À la figure 3.13, les arêtes en bleu sont candidates à l'arête précédente de l'arête en rouge qui est présentement en traitement. À partir du sommet C qui est le premier sommet de l'arête CE en traitement, quatre arêtes de contour ont été détectées. L'arête CE est exclue des candidates puisqu'il s'agit de l'arête en traitement. L'arête CF est aussi exclue puisqu'elle appartient au même triangle que l'arête en traitement. Les arêtes AC et DC sont toujours candidates et les nouveaux triangles ACE et CED seront évalués.

Les flèches à l'intérieur des triangles à la figure 3.13 indiquent le sens de définition des arêtes de ces triangles. Les flèches noires indiquent le sens des arêtes pour les triangles existants et les flèches vertes indiquent celui des arêtes des nouveaux triangles possibles formés avec les arêtes en traits pointillés. Le sens des arêtes des nouveaux triangles dépend du sens de l'arête en traitement dans le triangle de contour existant. L'arête en traitement à la figure 3.13 est définie à partir du sommet E vers le sommet C pour le triangle CFE . Le sens de cette arête doit nécessairement être inversé pour les nouveaux triangles afin de conserver la continuité de la surface sinon les vecteurs normaux des

triangles seront inversés. L'arête en traitement est donc définie à partir du sommet C vers le sommet E pour les nouveaux triangles. Les deux autres arêtes des triangles doivent suivre le sens de l'arête CE pour que toutes les arêtes soient définies dans le même sens. Les deux nouveaux triangles possibles sont donc les triangles ACE et CED .

Le test de continuité sur ces triangles consiste à vérifier le sens de l'arête précédente dans le triangle existant et dans le nouveau triangle. Si le sens de l'arête est inversé dans les deux triangles alors la surface est continue, sinon elle est inversée. Par exemple, si le nouveau triangle ACE réussit le test de la sphère alors le test de continuité sera appliqué sur celui-ci. Le sens de l'arête précédente AC à la figure 3.13 qui est commune au triangle ABC et au nouveau triangle ACE est inversé. La surface est donc continue pour ce nouveau triangle et il sera ajouté à la surface existante. Dans le second cas possible, si le nouveau triangle CED réussit le test de la sphère alors le test de continuité sera appliqué sur celui-ci. Le sens de l'autre arête précédente DC à la figure 3.13 qui est commune au triangle BDC et au nouveau triangle CED est le même pour les deux triangles. La surface n'est donc pas continue dans ce cas-ci et ce triangle ne sera pas ajouté à la surface.

Afin de déterminer si un triangle est un triangle de contour dans le dernier des trois cas possibles (figure 3.4D) pour l'ajout d'un nouveau triangle, il suffit de vérifier si le triangle contient au moins une arête de contour. Si c'est le cas alors c'est un triangle de contour. Un triangle de contour peut aussi contenir deux arêtes de contour. Dans la recherche du sommet le plus près de l'arête de contour en traitement pour un triangle de contour, seulement les sommets des arêtes de contour du triangle sont considérés. Si le triangle possède une seule arête de contour alors deux sommets sont considérés et s'il en possède deux alors les trois sommets le sont. Pour déterminer le sommet de contour le plus près de l'arête en traitement, la somme des distances entre chacun des sommets de l'arête en traitement et le sommet de contour du triangle est calculée et le sommet du

triangle pour lequel cette distance cumulée est la plus petite est sélectionné. La figure 3.14 présente le principe du sommet de contour le plus près de l'arête en traitement.

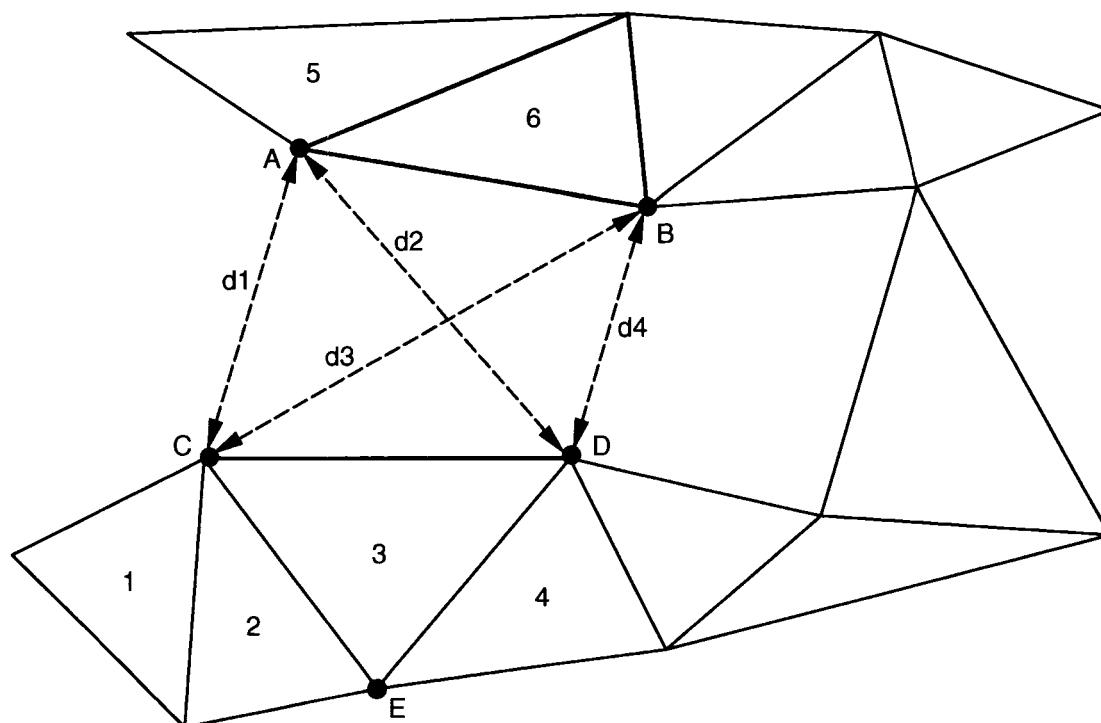


Figure 3.14 Sommet de contour le plus près de l'arête en traitement

À la figure 3.14, l'arête en traitement est en bleu et le triangle de contour de même orientation que celui de l'arête en traitement et qui interférerait avec la sphère du nouveau triangle original est en rouge. Le triangle 6 possède une seule arête de contour, seulement les deux sommets de cette arête sont considérés. Les distances $d1$ à $d4$ sont calculées. Si la somme de $d1$ et $d2$ est plus petite que la somme de $d3$ et $d4$, alors le sommet A devient le troisième sommet du nouveau triangle. Dans le cas contraire, c'est le sommet B qui est conservé. Plutôt que le triangle 6, si le triangle 5 était celui sélectionné pour déterminer le troisième sommet du nouveau triangle, alors les trois sommets de ce triangle seraient considérés dans le calcul de la distance minimale puisque ce triangle possède deux arêtes de contour.

Avant de considérer le nouveau triangle formé avec le troisième sommet sélectionné, il faut vérifier si le troisième sommet est bien différent des deux autres et s'il n'appartient pas au même triangle que celui constitué de l'arête en traitement. Cette vérification supplémentaire est nécessaire car un triangle précédent ou suivant à la figure 3.14 relié par un seul point (triangle 1) ou par une arête (triangle 2) au triangle constitué de l'arête en traitement (triangle 3) pourrait aussi bien répondre aux critères de sélection pour ce cas qu'un triangle de contour de même orientation d'une autre partie non voisine de la surface.

Si cette situation se présente avec le triangle 1, alors le sommet du triangle 1 le plus près de l'arête en traitement est évidemment le sommet *C* qui appartient aussi à l'arête en traitement. Dans ce cas le nouveau triangle ne serait pas cohérent puisqu'il aurait deux sommets identiques. Si cette situation se présente avec le triangle 2, alors le sommet de contour du triangle 2 le plus près de l'arête en traitement est le sommet *E* qui appartient aussi au triangle 3. Dans ce cas le nouveau triangle ne conviendrait pas car il serait identique au triangle 3. Dans un cas comme dans l'autre, si le sommet ne convient pas alors le triangle est rejeté. S'il convient et que le triangle réussit le test de la sphère alors le test supplémentaire de continuité de la surface décrit précédemment doit être fait avant d'ajouter le triangle à la surface. Ce dernier test sert à éliminer des cas similaires à celui présenté à la figure 3.13.

3.1.3 Améliorations à l'algorithme

D'après les tests effectués sur différentes surfaces, l'algorithme de triangulation « marching triangle » fonctionne seulement sur des surfaces de synthèse planes comme celle présentée à la figure 3.6 étant donné que ce type de surfaces représente un cas particulier relativement trivial. Plusieurs améliorations ont été apportées à l'algorithme pour tenter de produire un maillage satisfaisant sur des surfaces quelconques afin de le rendre fonctionnel et efficace. Ces améliorations sont décrites aux sections suivantes.

3.1.3.1 Test de la sphère

Le principal problème de l'algorithme est le test de la sphère pour les nouveaux triangles à ajouter à la surface. La contrainte de surface de ce test est trop grande, ce qui résulte que peu de triangles sont ajoutés à la surface résultante. Ainsi le résultat final contient seulement une portion de la surface ou encore l'ensemble de la surface parsemé de plusieurs trous. Le but du test de la sphère est d'éviter qu'un nouveau triangle entre en conflit avec des triangles existants. Il arrive souvent que le nouveau triangle n'interfère pas avec le reste de la surface. Celui-ci pourrait être ajouté sans problème mais le test de la sphère échoue car des triangles sont à l'intérieur de la sphère sans toutefois être en conflit avec le nouveau triangle. La figure 3.15 illustre ce cas problématique.

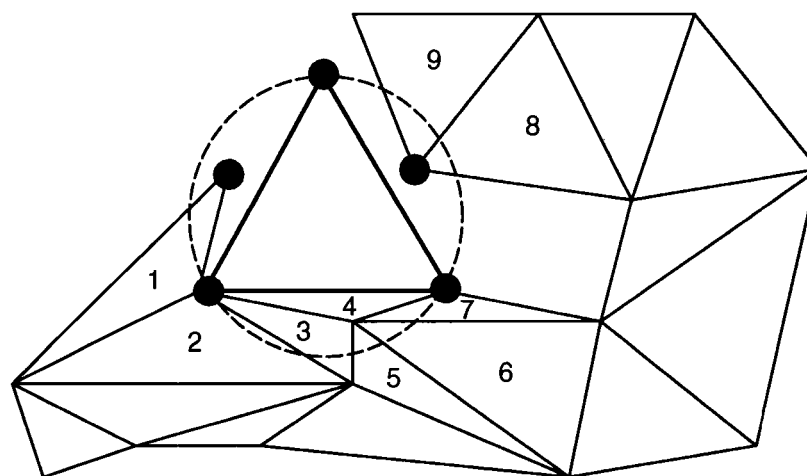


Figure 3.15 Cas problématique du test de la sphère

À la figure 3.15, les neuf triangles numérotés sont à l'intérieur de la sphère. Le triangle 4 est exclu du test puisqu'il est constitué de l'arête en traitement. Il reste cependant huit triangles problématiques et pourtant le nouveau triangle pourrait être ajouté à la surface. Le cas illustré est exagéré pour montrer les différents cas possiblement rencontrés mais un seul de ces huit triangles est suffisant pour faire échouer le test. Dans certains cas, il

y a un intérêt à faire échouer le test initial même si le nouveau triangle pouvait être ajouté. À la figure 3.15 par exemple, si le test initial réussit alors le petit espace entre le nouveau triangle et le triangle 1 ne sera probablement jamais comblé par aucun triangle, laissant ainsi un trou sur la surface. Cet espace ne sera pas comblé parce que le test de la sphère sur le triangle formé de l'arête rouge près du triangle 1 et du sommet vert du triangle 1 ne réussira pas.

Il serait préférable que le test de la sphère du nouveau triangle échoue au départ afin de tester et d'ajouter un triangle plus cohérent à la surface. Dans le contexte de la figure 3.15, deux triangles plus cohérents seraient ceux formés de l'arête bleue et de l'un ou l'autre des sommets verts. Ces triangles seraient plus cohérents puisqu'ils ne laisseraient pas d'espace entre les triangles de la surface. Cependant, les tests de la sphère pour ces deux nouveaux triangles ne réussiront pas plus que le test du nouveau triangle de départ puisqu'il y aura toujours quelques triangles à l'intérieur des sphères. Finalement aucun triangle ne sera ajouté à l'arête en traitement.

Le test de la sphère a été modifié en remplaçant la sphère originale par une autre sphère plus petite qui diminue la contrainte de surface du test. Cette sphère passe par le nouveau sommet du triangle et par le point milieu de l'arête en traitement, son diamètre est égal à la distance entre ces deux points. Le centre de cette sphère est le point milieu entre le nouveau sommet et le point milieu de l'arête en traitement et son rayon est égal à la distance entre son centre et le nouveau sommet. La figure 3.16 présente la nouvelle sphère de test.

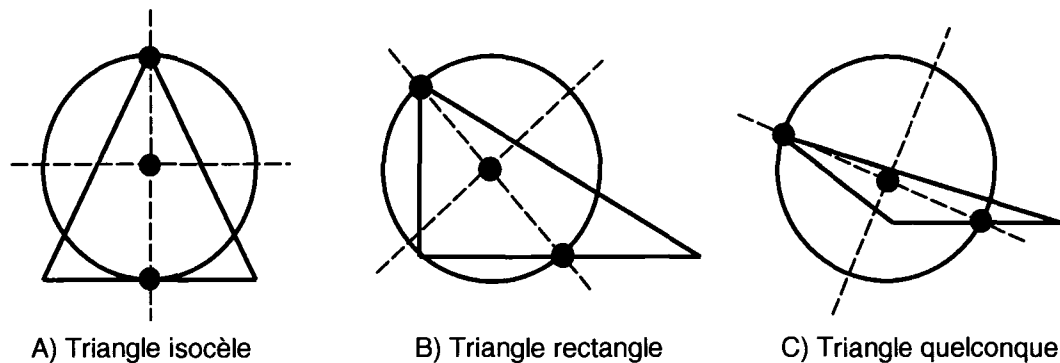


Figure 3.16 Nouvelle sphère pour le test

À la figure 3.16, la nouvelle sphère est illustrée pour trois triangles différents. L'arête en traitement est en bleu, le troisième sommet et les nouvelles arêtes sont en vert. La sphère en question et son centre sont en rouge. Cette sphère n'est pas encore parfaite pour l'application mais les résultats obtenus sont grandement améliorés avec celle-ci comparativement à celle originale. Avec l'implantation de ce nouveau test, il est maintenant possible de trianguler adéquatement la plupart des surfaces de synthèse continues qui sont définies mathématiquement. De plus, le calcul du centre de cette nouvelle sphère est beaucoup plus simple et rapide que celle circonscrite au triangle. Il suffit de calculer le point milieu de l'arête en traitement en effectuant la moyenne de ses sommets et ensuite de calculer le point milieu entre le troisième sommet et le point milieu de l'arête en effectuant une deuxième moyenne sur ces points. Pour des surfaces réelles qui proviennent du MapScan, les résultats sont meilleurs qu'avec la sphère originale mais il reste tout de même plusieurs trous sur la surface.

Pour améliorer davantage les résultats, un autre objet de test a été conçu pour remplacer la sphère. Il s'agit d'un triangle 3D tel qu'illustré à la figure 3.17. Cette solution potentielle n'a pas été implantée et testée dans le cadre du présent mémoire.

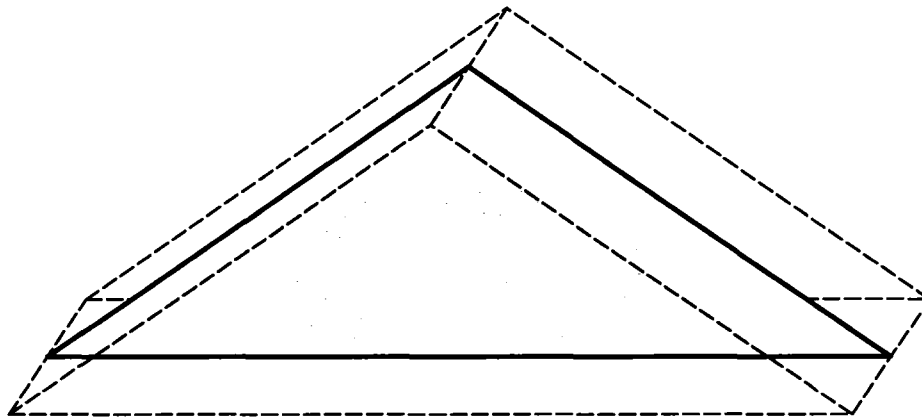


Figure 3.17 Nouvel objet potentiel pour le test

À la figure 3.17, le nouveau triangle est au centre du prisme. Le seul paramètre à déterminer pour définir ce prisme est son épaisseur et elle pourrait être déterminée en fonction de la résolution de la grille utilisée pour la fonction de champ. Pour implanter ce test il faudrait vérifier pour chacun des triangles de la surface s'il y a une intersection commune avec chacun des côtés du prisme. Même s'il n'y a pas d'intersection, il faudrait ajouter un test supplémentaire pour s'assurer que le triangle n'est pas complètement à l'intérieur du prisme en vérifiant que le triangle est du même côté des deux plans formés par les triangles devant et derrière le nouveau triangle selon le point de vue de la figure 3.17. Le nouveau test avec le triangle 3D serait cependant plus complexe à implanter. Le temps de calcul serait plus grand mais l'algorithme serait peut-être plus efficace.

3.1.3.2 Ordre de traitement des arêtes

L'ordre de traitement des arêtes est aléatoire du point de vue du contour de la surface. Lorsqu'un nouveau triangle est ajouté à la surface, les nouvelles arêtes sont placées à la fin de la liste des arêtes dans la structure de données qui caractérise la surface. Un test a été effectué pour vérifier si cet ordre avait un impact sur le résultat. Pour une même surface, l'algorithme a été testé à deux reprises en modifiant l'ordre de traitement des

arêtes entre les deux triangulations et les résultats ont été comparés. Les résultats sont différents ce qui signifie que cet algorithme est dépendant de l'ordre de traitement des arêtes. Ce fait s'explique en observant le fonctionnement de l'algorithme. Pour une même arête de contour, lorsque l'algorithme arrive à cette arête, la surface n'est pas nécessairement la même et dans un cas le nouveau triangle peut être ajouté alors que dans l'autre il peut être rejeté. Dans le second cas, un autre triangle peut être ajouté ou non en fonction de la configuration des triangles voisins. La figure 3.18 présente un cas possible où la surface serait modifiée par l'ordre de traitement des arêtes.

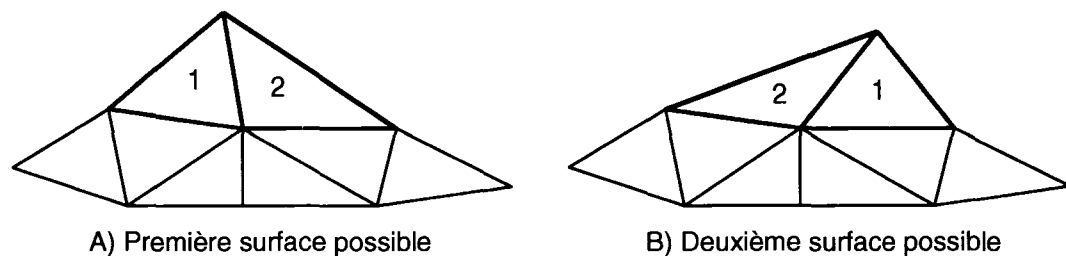


Figure 3.18 Surface modifiée par l'ordre de traitement des arêtes

À la figure 3.18, la surface est supposée identique avant l'ajout des triangles 1 et 2. Dans les deux cas, l'arête bleue est traitée en premier et l'arête rouge en deuxième. L'ordre de traitement est donc modifié et inversé pour ces arêtes. Le triangle 1 est ajouté en premier et le triangle 2 en deuxième. Le triangle 1 représente un nouveau triangle à partir du nouveau point original et le triangle 2 représente un nouveau triangle relié à l'arête précédente dans le premier cas (figure 3.18A) et relié à l'arête suivante dans le second (figure 3.18B). Les deux surfaces sont devenues différentes et les nouveaux triangles qui s'ajouteront aux nouvelles arêtes contribueront à rendre les surfaces encore plus différentes.

Le fait que l'algorithme soit dépendant de l'ordre de traitement des arêtes n'est pas souhaitable car c'est un paramètre supplémentaire qui doit être pris en considération.

Jusqu'à présent l'ordre de traitement était aléatoire mais il existe possiblement un ordre qui augmenterait l'efficacité de l'algorithme. Intuitivement, l'ordre le plus plausible serait sûrement l'ordre qui suit le contour de la surface. Cet essai n'a cependant pas été réalisé dans le cadre du présent mémoire. Si cette modification doit être implantée, il faut préciser davantage l'ordre de traitement. Au départ la recherche d'une première arête de contour doit être faite et à partir de cette arête arbitraire, il faut suivre le contour de la surface en effectuant pour chaque nouvelle arête une recherche de l'arête de contour voisine dans un des deux sens. Selon l'arête arbitraire de départ choisie, le résultat est susceptible d'être différent d'une arête à l'autre. Selon le sens choisi pour suivre le contour, le résultat est encore susceptible d'être différent.

Une fois qu'un nouveau triangle a été ajouté à une arête de contour, il y a plusieurs possibilités pour poursuivre l'algorithme. Deux de ces possibilités semblent intuitivement intéressantes. La première pourrait traiter immédiatement et récursivement les nouvelles arêtes ajoutées à la surface jusqu'au retour à l'arête qui suit l'arête de départ dans le contour original. La seconde pourrait poursuivre le contour original jusqu'à la fin de la boucle et ensuite recommencer le nouveau contour. Si cette seconde option est choisie alors un autre choix s'impose sur le fait de traiter ou non les arêtes de contour qui étaient présentes dans le contour original. L'algorithme original spécifie que toutes les arêtes doivent être traitées une seule fois.

À ce sujet, un test a été réalisé sur l'algorithme original en traitant les arêtes dans l'ordre aléatoire qu'elles se présentent dans la liste et en ajoutant les nouvelles arêtes à la fin comme auparavant. Lorsque l'algorithme s'arrête suite au traitement de la dernière arête, il est relancé sur la surface résultante et de nouveaux triangles sont ajoutés à la surface lors de la seconde itération. Un processus itératif a donc été implanté. Lorsque l'algorithme s'arrête après la dernière arête dans la liste, il est systématiquement relancé tant et aussi longtemps qu'il y a eu au moins un nouveau triangle ajouté à l'itération précédente. Le résultat final est encore loin d'être parfait

mais il est supérieur au résultat après seulement une itération. Après quelques essais sur des surfaces choisies arbitrairement, le nombre d'itérations nécessaires avant l'arrêt définitif de l'algorithme a varié entre 3 et 14 itérations.

3.1.3.3 Nouveau triangle possible

Dans le dernier cas possible (figure 3.4D) présenté pour l'ajout d'un nouveau triangle et détaillée au dernier paragraphe de la section 3.1.2 de ce document, seulement le sommet de contour qui est le plus près de l'arête en traitement est utilisé pour former un nouveau triangle. Il se peut dans certains cas que le sommet le plus près ne convienne pas et que le triangle ne soit pas ajouté mais qu'un autre sommet de contour du triangle aurait convenu et que ce nouveau triangle puisse être ajouté à la surface. L'algorithme a donc été modifié et tous les sommets de contour du triangle plutôt que seulement le plus près sont testés. Suite à cette modification quelques triangles supplémentaires ont été ajoutés pour une surface donnée. La figure 3.19 illustre une des possibilités pour ce cas.

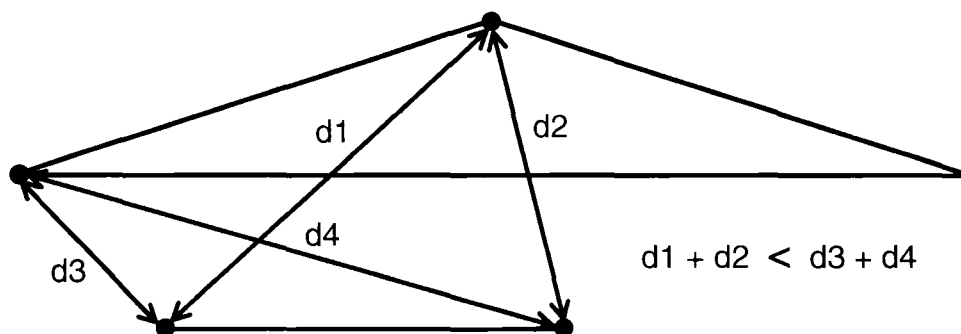


Figure 3.19 Vérification de tous les sommets de contour

À la figure 3.19, l'arête en traitement est représentée en bleu. Le sommet du triangle de contour en noir qui est le plus près de cette arête est celui en rouge. Ce nouveau triangle formé avec les arêtes en rouge ne convient pas à la surface car il interfère avec le triangle de contour. Le sommet en vert est plus éloigné de l'arête que celui en rouge

mais le nouveau triangle formé de ce sommet et des arêtes en vert convient à la surface car il n'interfère pas avec le triangle de contour. Ce nouveau triangle possible n'était pas évalué selon l'algorithme original.

3.1.3.4 Efficacité de l'algorithme amélioré

À partir de la version la plus efficace implanté jusqu'à présent de l'algorithme du « marching triangle », plusieurs essais ont été réalisés sur différentes surfaces. L'algorithme fonctionne parfaitement bien sur des surfaces de synthèse. Si une surface est définie mathématiquement et qu'une fonction de champ est générée à partir de cette définition alors l'algorithme arrivera à trianguler adéquatement cette surface. Pour obtenir un algorithme qui fonctionne correctement, il faut partir d'une surface continue sans discontinuité ni trou, bruit ou autre imperfection. Les surfaces réelles provenant d'un capteur 3D tel que le MapScan sont bruitées et elles contiennent des imperfections de surfaces, des trous et des discontinuités. Lorsque l'algorithme rencontre une discontinuité par exemple, il s'arrête à cet endroit et il n'est pas en mesure de continuer à trianguler la surface dans cette direction. De plus, plusieurs triangles qui devraient être ajoutés à la surface ne le sont pas et le modèle final est toujours parsemé de trous.

Cet algorithme dans sa version actuelle est donc inapproprié pour les surfaces réelles du MapScan. Malgré toutes les améliorations apportées à l'algorithme du « marching triangle » original pour le rendre plus efficace et l'adapter aux données du MapScan, il a tout de même été abandonné. L'algorithme adapté est plus performant que celui de base mais les résultats obtenus jusqu'à présent ne sont pas suffisamment satisfaisant pour l'employer avec le logiciel du MapScan. Un autre algorithme de triangulation a été utilisé. Il s'agit de l'algorithme du « marching cube » qui a été implanté et adapté aux données des fonctions de champ. Le fonctionnement de cet algorithme est décrit à la section suivante de ce document.

Plusieurs raisons avaient motivé la décision d'utiliser le « marching triangle » comme algorithme pour la triangulation du modèle final. Tout d'abord cet algorithme a été développé par les mêmes auteurs qui ont conçu la méthode de fusion des surfaces par la représentation implicite. Au moment de choisir la méthode de triangulation, le calcul des fonctions de champs ainsi que l'intégration de celles-ci avaient été implantés avec succès. Cette dernière étape du projet semblait adaptable aux données du MapScan comme les précédentes. Ensuite il est mentionné dans la littérature que la méthode globale (représentation implicite et triangulation) a été testée avec succès sur des données provenant d'un capteur 3D qui sont similaires à celles du MapScan. Cette méthode de triangulation est aussi très bien détaillée et des résultats intéressants sont présentés.

Finalement les auteurs du « marching triangle » comparent avantageusement cet algorithme à celui du « marching cube » sur différents points dans la littérature. Pour une surface de résolution équivalente, le « marching triangle » produirait une surface composée de moins de triangles qui seraient plus uniformes en grandeur que le « marching cube ». La représentation graphique du modèle final serait donc de meilleure qualité et une plus petite quantité de mémoire serait nécessaire pour le sauvegarder. Le temps de calcul du « marching triangle » serait aussi plus court que celui du « marching cube ». Pour les deux paramètres, la quantité de mémoire et le temps de calcul, un gain d'un facteur trois à cinq serait réalisé en optant pour le « marching triangle ».

Les algorithmes de triangulation « marching triangle » et « marching cube » sont de natures très différentes. Ils n'ont pratiquement aucun point en commun à l'exception du fait qu'ils sont tous les deux en mesure de produire des triangles à partir d'une fonction de champ pour obtenir une surface polygonale. Les problèmes rencontrés avec le « marching triangle » comme l'arrêt de la triangulation aux discontinuités et les surfaces résultantes parsemées de trous ne sont pas présents sur les surfaces produites par le

« marching cube » puisque ces problèmes sont directement liés à la nature de l'algorithme du « marching triangle ».

3.2 Triangulation « marching cube »

Le « marching cube » est un algorithme de triangulation de type volumétrique parce qu'il s'exécute sur des volumes de base indépendamment du reste de la surface. Pour chacun des volumes de base symbolisés par des cubes, l'algorithme génère des triangles à l'intérieur de ce volume. Lorsque le traitement d'un cube est terminé, le suivant est entrepris et ainsi de suite de façon à couvrir l'ensemble du volume global de la surface implicite intégrée. Cet algorithme est utilisé pour produire le maillage de la surface finale. La définition de l'algorithme, son implémentation informatique ainsi que les améliorations qui y ont été apportées sont détaillées aux sections suivantes.

3.2.1 Algorithme du « marching cube »

L'algorithme du « marching cube » s'exécute sur des volumes cubiques de base appelés voxels. Chaque voxel, qui est un cube élémentaire, possède huit sommets contenant une information de distance à la surface. L'algorithme est exécuté sur un voxel à la fois et il génère un ou des triangles à l'intérieur du voxel selon les informations de distance aux sommets du voxel. Une fois que tous les voxels ont été traités, la construction de la surface finale est terminée. Pour un voxel en particulier, il existe plusieurs configurations possibles pour générer les triangles. Si les huit sommets ont une distance positive ou encore s'ils ont tous une distance négative, ceci signifie que le voxel est complètement au-dessus de la surface dans le premier cas ou complètement en dessous dans le second et que la surface ne passe pas par ce voxel. Aucun triangle ne sera généré à l'intérieur de ce voxel. S'il y a des distances positives et négatives dans les huit sommets d'un voxel, ceci signifie que la surface passe à l'intérieur de ce voxel et un ou des triangles seront générés à l'intérieur du voxel selon la configuration des signes. La figure 3.20 présente quelques cas possibles de triangulation des voxels.

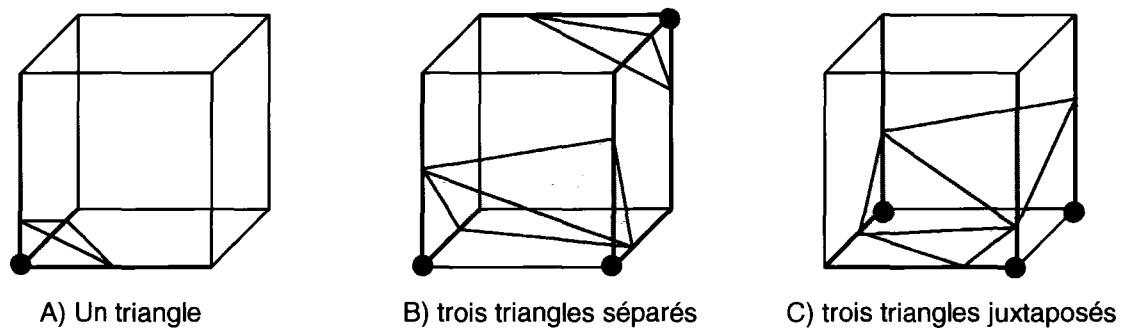


Figure 3.20 Triangulation des voxels

À la figure 3.20, les sommets des voxels ayant des distances positives sont identifiés par les sphères en bleu. La première étape de la triangulation est d'étiqueter les sommets selon le signe de la distance. Par exemple, à la figure 3.20A seulement un sommet a une distance positive, il s'agit du coin inférieur avant gauche. La seconde étape est d'identifier les arêtes qui sont bornées par des sommets ayant des distances de signe opposé. Ces arêtes sont identifiées en rouge à la figure 3.20. Par exemple, à la figure 3.20A trois arêtes répondent à ce critère et ce sont celles qui touchent au seul sommet positif. La troisième étape est de calculer par interpolation linéaire le point par lequel la surface passe sur chacune des arêtes. Il s'agit de trouver le zéro de la fonction qui correspond à l'endroit exact de traversée de la surface. La quatrième étape est de relier ces points ensemble pour former les triangles de la surface à l'intérieur de ce voxel. S'il y a seulement trois points, comme c'est le cas à la figure 3.20A, alors seulement un triangle est généré. Quatre points reliés entre eux donneront naissance à deux triangles et cinq points à trois triangles comme c'est le cas à la figure 3.20C.

Selon la configuration des signes, il est possible que tous les triangles soient juxtaposés comme à la figure 3.20C ou encore qu'ils soient séparés comme à la figure 3.20B. Ce dernier cas possède sept points interpolés et seulement trois triangles étant donné que les triangles sont séparés. Plusieurs sommets de même signe qui sont tous directement reliés par des arêtes sans passer par un sommet de signe opposé généreront un groupe

de triangles juxtaposés. S'il existe un autre groupe de sommets tel que décrit précédemment dans le même voxel et isolé du groupe précédent alors un autre groupe de triangles juxtaposés et isolés des précédents seront générés. Un groupe de sommets peut aussi être réduit à un seul comme c'est le cas à la figure 3.20B.

Comme un voxel possède huit sommets et que chacun d'eux peut avoir deux états (signe positif ou négatif), il existe 256 configurations différentes pour la génération des triangles. Cependant plusieurs cas similaires peuvent être traités de la même façon. Par exemple, le cas à la figure 3.20A résume seize cas différents. Le sommet positif peut être n'importe lequel des huit et le traitement sera le même. De plus, le sommet isolé des autres pourrait aussi être un sommet négatif plutôt qu'un positif. La seule différence avec le changement de signe est que le sens du vecteur normal au triangle sera opposé. Lorsque la triangulation d'un voxel est terminée, le voxel suivant est traité. Deux voxels voisins ont quatre sommets et quatre arêtes en commun. Pour ces entités communes, la même configuration sera étiquetée et les mêmes points interpolés seront calculés. Les triangles ayant des sommets sur la frontière entre deux voxels voisins seront nécessairement juxtaposés et ceci assure la continuité de la surface.

3.2.2 Implémentation informatique

Comme le « marching cube » fonctionne à partir de voxels, il faut faire une analogie avec la grille 3D de la fonction de champ intégrée. En prenant huit éléments voisins en forme de cube de la fonction de champ, la configuration de voxels est obtenue. Pour la triangulation, la fonction de champ est donc perçue comme un ensemble de voxels qui sont parcourus un à la suite des autres pour générer les triangles. La grandeur des triangles dans le modèle final est directement liée à la résolution de la grille de la fonction de champ. La résolution fixe la distance entre les éléments et nécessairement la grandeur des voxels à l'intérieur desquels les triangles sont générés. Si la grandeur des voxels est modifiée, la grandeur des triangles sera aussi modifiée proportionnellement à celle des voxels.

La classe « MarchingCubesTriangulator » a été développée pour réaliser la triangulation de la fonction de champ. Pour accélérer et simplifier le traitement, tout le processus de triangulation fonctionne à l'aide de deux tables de correspondances définies dans cette classe. Ces tables font références à des indices de sommets et d'arêtes du voxel. La convention établie est présentée à la figure 3.21A. Les indices des sommets sont en bleu et ceux des arêtes en rouge.

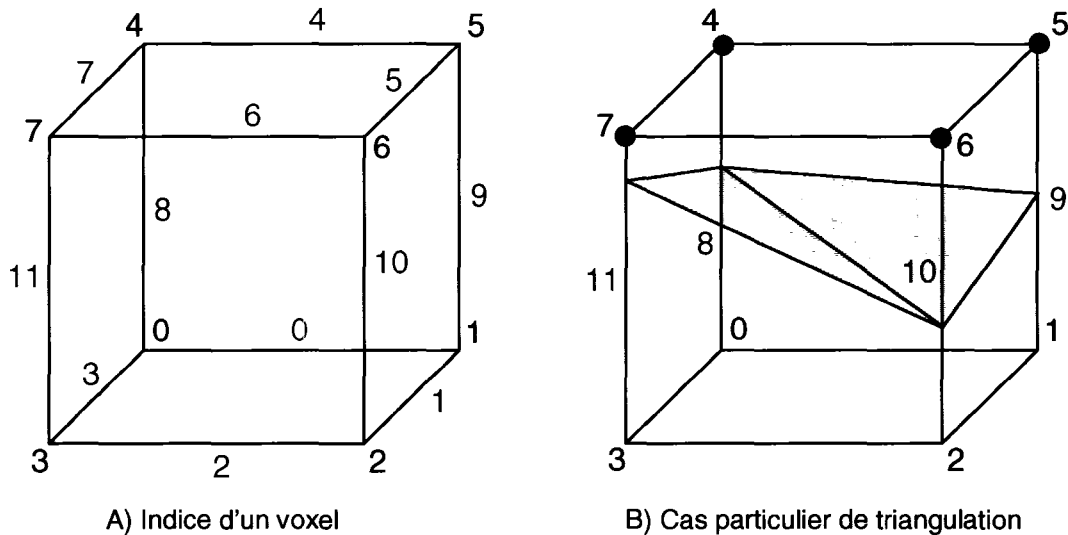


Figure 3.21 Convention des indices d'un voxel et exemple de triangulation

La figure 3.21B présente un cas particulier de triangulation d'un voxel pour visualiser les explications sur la nature des tables de correspondances et l'implémentation de l'algorithme du « marching cube ». Les sphères vertes à la figure 3.21B représentent des distances positives. Pour qu'un voxel soit triangulé, il faut que ses huit sommets soient considérés comme des éléments utiles à la fonction de champ. Si au moins un élément du voxel est étiqueté comme étant non utile à la fonction de champ alors ce voxel est ignoré lors du processus de triangulation.

Un code de 8 bits est construit avec le signe des distances à chacun des sommets du voxel. Le poids du sommet dépend de son indice dans le voxel et l'indice 0 est le moins

significatif. Pour un sommet, si la distance est négative alors un [1] est placé dans le code de 8 bits à l'endroit indiqué par l'indice du sommet. À la figure 3.21B, le code de 8 bits est [0000 1111] et la valeur décimale correspondante est 15. La première table de correspondance contient 256 données de 12 bits. Chacune des données représente une configuration possible pour la triangulation d'un voxel. Le code de 12 bits d'une donnée détermine les arêtes qui doivent être interpolées pour cette configuration de signe. Dans le cas à la figure 3.21B, la 15^e donnée de la première table est récupérée et le code de 12 bits est [1111 0000 0000]. Chacune des bits de ce code correspond à une arête et l'arête ayant l'indice 0 est la moins significative. Les [1] du code indique qu'il faut interpoler sur ces arêtes pour obtenir les sommets des triangles. Donc les arêtes 8, 9, 10 et 11 seront interpolées dans ce cas-ci. Ensuite les interpolations sont effectuées à l'aide de l'équation 3.11 pour obtenir les coordonnées des sommets des éventuels triangles.

$$P' = P_1 + \left(\frac{d_1}{d_1 - d_2} \right) (P_2 - P_1) \quad (3.11)$$

À l'équation 3.11, le point P' représente le point interpolé à partir des points P_1 et P_2 . Les variables d_1 et d_2 représentent les distances à la surface aux points P_1 et P_2 respectivement. Le calcul est effectué pour les trois composantes (x,y,z) du point P' .

La deuxième table de correspondance possède deux dimensions et elle contient 256 x 16 données. La première dimension de 256 sert encore à représenter toutes les configurations possibles de triangulation d'un voxel. Pour un cas particulier, la seconde dimension sert à déterminer l'ordre dans lequel il faut relier les sommets interpolés pour construire les bons triangles. Ces 16 données représentent les indices des arêtes par groupe de trois. Par exemple, dans le cas présent ces seize données sont [(9, 8, 10); (10, 8, 11); (-1, -1, -1); (-1, -1, -1); (-1, -1, -1); -1]. Ceci signifie que les sommets du premier triangle généré sont les points interpolés sur les arêtes 9, 8 et 10 et ceux du

second sont 10, 8 et 11. L'ordre des trois sommets pour un même triangle est important pour obtenir le bon sens du vecteur normal au triangle. Celui-ci est défini à l'aide de la règle de la main droite en suivant l'ordre des indices dans le groupe de trois. La figure 3.22 illustre cette règle.

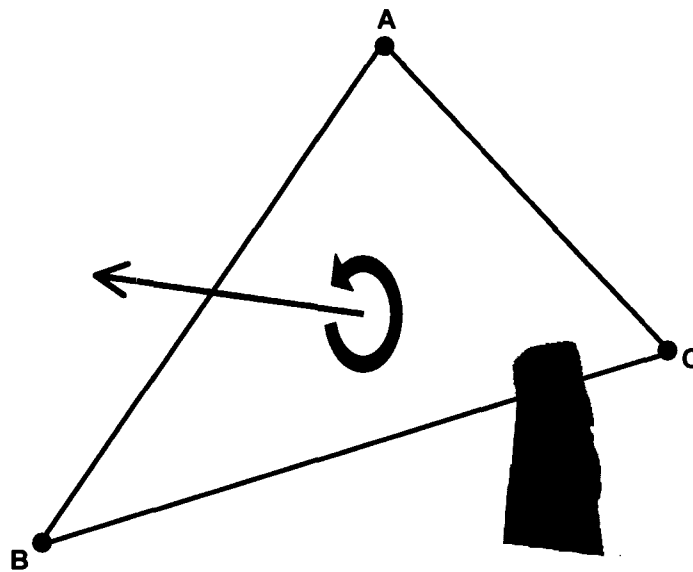


Figure 3.22 Règle de la main droite pour l'orientation du vecteur normal

À la figure 3.22, le triangle est défini par ses trois sommets dans l'ordre A , B et C . La flèche bleue indique une rotation dans le plan du triangle dans l'ordre de définition de ses sommets. En courbant les doigts de la main droite de façon à ce qu'ils suivent le sens de rotation de la flèche bleue, le pouce indique le sens du vecteur normal au triangle. Ce vecteur est représenté à la figure 3.22 par la flèche rouge. En inversant l'ordre de définition des sommets du triangle (A, C, B), le vecteur normal serait plutôt en sens opposé à celui indiqué par la flèche rouge.

Cinq triangles au plus peuvent être générés dans un voxel selon la configuration des signes. Dans le cas à la figure 3.21B, seulement deux triangles sont générés. Les indices (-1) dans la seconde table de correspondance signifient qu'il n'y a pas d'autre triangle à

généraliser. L'algorithme de génération des triangles cesse de récupérer les données par groupe de trois lorsqu'il rencontre l'indice (-1). Le dernier indice (-1) des seize données sert de critère d'arrêt au générateur de triangle lorsque le cas le plus complexe où cinq triangles sont générés est rencontré. Lorsque tous les voxels ont été triangulés, la surface finale est complète. Cette surface polygonale explicite peut maintenant être sauvegardée et convertie à un format compatible avec d'autres logiciels. Les étapes de la triangulation pour chaque voxel se résument de la façon suivante :

1. Vérifier que les huit sommets du voxel sont utiles à la fonction de champ. Sinon aucun triangle n'est créé. Fin de la triangulation pour ce voxel.
2. Générer le code binaire de 8 bits en fonction des signes des distances aux sommets.
3. Si le code correspond au nombre décimal 0 ou 255, alors aucun triangle n'est créé. Fin de la triangulation pour ce voxel.
4. Récupérer le code binaire de 12 bits dans la première table de correspondance selon le code binaire de 8 bits généré.
5. Calculer les coordonnées des sommets des triangles par interpolation sur certaines arêtes du voxel selon le code binaire de 12 bits.
6. Créer les triangles avec les sommets interpolés dans l'ordre établi selon la seconde table de correspondance.
7. Ajouter les triangles à la surface résultante. Fin de la triangulation pour ce voxel.

3.2.3 Améliorations à l'algorithme

L'algorithme du « marching cube » tel que décrit à la section précédente présente certaines failles. À l'occasion des triangles incohérents ou qui n'appartiennent pas à la surface sont ajoutés. Ces deux situations ainsi que les modifications apportées à l'algorithme pour les corriger sont présentées aux deux sections suivantes.

3.2.3.1 Triangles incohérents

Il arrive que des triangles incohérents soient ajoutés à la surface. Cette situation se présente lors des cas limites où la distance à un sommet est identiquement nulle. Par exemple, à la figure 3.20A, si le seul sommet positif possède une distance égale à zéro, alors les trois points interpolés auront exactement les mêmes coordonnées qui seront celles du sommet du voxel. Le triangle engendré par ces trois points est donc incohérent puisqu'il s'agit du même point dans l'espace. Un autre exemple est celui de la figure 3.20C. Si les trois sommets positifs ont tous une distance nulle alors un seul des trois triangles est cohérent. Les deux autres triangles ont seulement deux sommets distincts, le troisième étant commun à un des deux autres. Ces cas limites ne sont pas traités dans l'algorithme de base. Une condition supplémentaire a été ajoutée pour vérifier que les trois sommets d'un triangle ont bien des coordonnées différentes avant d'ajouter ce triangle à la surface.

3.2.3.2 Triangles supplémentaires

Il arrive que des triangles supplémentaires qui n'appartiennent pas à la surface soient ajoutés à celle-ci. Ces triangles isolés se retrouvent au-dessus ou en dessous de la surface à une certaine distance de celle-ci sans toutefois en faire partie. Cette situation se présente lorsqu'un voxel en traitement possède des sommets positifs et négatifs et que les distances de ces sommets sont plus grandes que la résolution de la grille de la fonction de champ. Il est possible, en présence d'une surface d'une certaine complexité, d'obtenir deux éléments voisins d'une fonction de champ et qu'un des éléments possède une distance positive alors que l'autre possède une distance négative sans que la surface passe réellement entre ces deux éléments. Ce cas est rencontré si les points les plus près de la surface pour ces éléments proviennent de deux triangles ayant des orientations opposées. La figure 3.23 illustre ce cas en 2D. Le même principe se généralise en 3D.

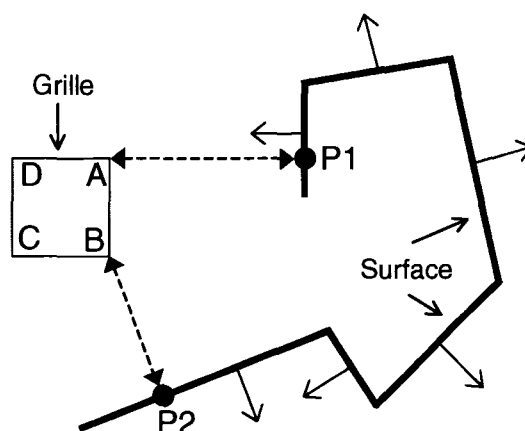


Figure 3.23 Éléments voisins de signe opposé

À la figure 3.23 un seul carré de la grille 2D est dessiné en bleu. Les vecteurs normaux aux différents segments de la surface sont affichés en rouge. Le point $P1$ de la surface est le plus près de l'élément A alors que le point $P2$ est le plus près de l'élément B . Selon l'orientation de la surface, l'élément A est devant celle-ci et sa distance est positive alors que l'élément B est derrière la surface et sa distance est négative. Lorsque le carré dessiné de la grille sera en traitement pour la génération des segments de droite, une opposition de signe sera détectée entre les éléments A et B et un segment de droite sera généré entre ces éléments. L'emplacement du segment de droite dépendra aussi des signes aux éléments C et D du carré. Donc un segment de droite ne faisant pas partie de la surface sera ajouté par erreur.

Lorsqu'un cas comme celui à la figure 3.23 se présente, les distances en valeur absolue aux éléments A et B sont pratiquement toujours plus grandes que la résolution de la grille. En effet, si le carré de la grille est déplacé vers la droite de façon à le rapprocher du point $P1$, la distance à l'élément A diminuera et éventuellement le point le plus près de l'élément B se trouvera sur le même segment de droite que le point $P1$. À ce moment la distance à l'élément B deviendra positive et le cas problématique ne sera plus présent.

Le même raisonnement peut être fait en déplaçant le carré de la grille vers le bas. Dans ce cas la distance à l'élément A deviendra négative et le cas problématique disparaîtra.

Le cas illustré à la figure 3.23 n'est pas fréquent puisque généralement la résolution de la grille est relativement petite par rapport à la grandeur des triangles et que seulement les éléments près de la surface sont étiquetés comme étant utiles à la fonction de champ. Le cas présenté n'est pas traité dans l'algorithme de base du « marching cube ». Une condition supplémentaire a été ajoutée pour vérifier les valeurs absolues des distances des deux éléments à l'étape de l'interpolation pour trouver le zéro. Si la surface passe effectivement par un voxel et qu'elle coupe une arête alors les valeurs absolues des distances positive et négative aux sommets de cette arête qui serviront à l'interpolation sont nécessairement plus petites ou égales à la résolution de la grille. Si les deux distances sont plus grandes que la résolution de la grille alors le voxel est ignoré et aucun triangle n'est généré à l'intérieur de ce voxel.

CHAPITRE 4

ANALYSE DES RÉSULTATS

Ce chapitre présente les résultats obtenus avec l'algorithme de fusion des surfaces développé dans le cadre de ce projet. Plusieurs tests ont été réalisés dans le but de valider les différents algorithmes implantés et quelques résultats représentatifs de ces tests sont présentés. Le montage expérimental utilisé pour numériser les surfaces est décrit. Les résultats de fusion sur des surfaces planes ainsi que sur des surfaces courbes avec et sans redondance sont présentés. L'analyse des surfaces est réalisée pour décrire quelques imperfections sur les résultats obtenus. L'erreur induite par l'algorithme est mesurée sur les surfaces. Les performances globales de l'algorithme sont finalement présentées.

4.1 Montage expérimental

Un montage expérimental a été réalisé pour numériser les surfaces des objets 3D. Lorsque les surfaces sont numérisées, l'algorithme de fusion s'exécute sur celles-ci pour produire le modèle final. La figure 4.1 présente une photographie du montage expérimental avec le MapScan en action.

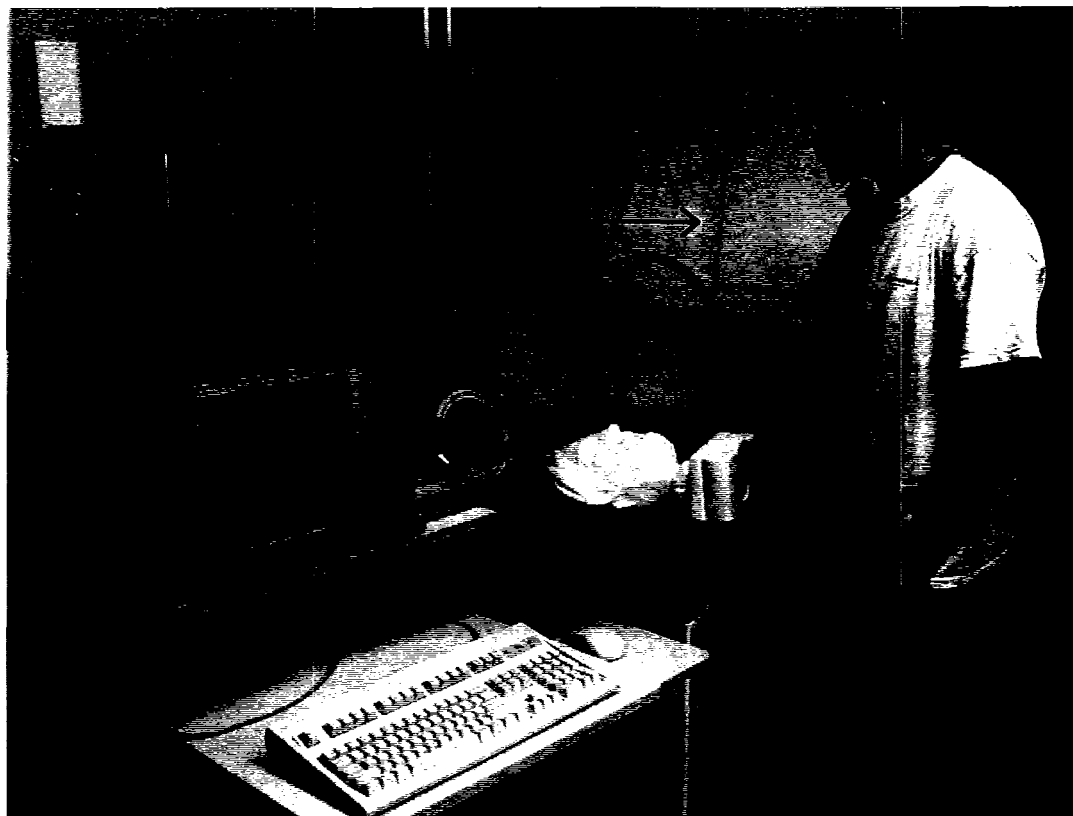


Figure 4.1 Photographie du montage expérimental

À la figure 4.1, la caméra 3D portable est tenue à la main par l'utilisateur. Celui-ci exécute un balayage sur la surface de l'objet 3D à numériser. Le profil laser suit la surface de l'objet. Le grand triangle avec les émetteurs est visible dans la partie supérieure, au centre de la photographie où il est fixé horizontalement. Le petit triangle avec les récepteurs est fixé à l'arrière du profilomètre. Ces composantes du système de positionnement sont pointées par les flèches bleues. L'utilisateur s'applique à tenir la caméra dans un angle où les deux triangles sont parallèles ainsi qu'à déplacer la caméra en translation seulement sur un seul axe indiqué par la flèche rouge. Cette procédure minimise les erreurs occasionnées par le système de positionnement.

Pour que le système fonctionne correctement, il ne doit pas y avoir d'obstruction entre les deux triangles. Les ondes ultrasonores doivent voyager librement entre les triangles. La distance entre les triangles doit respecter une plage d'opération d'environ 40 à 140 centimètres. Il ne doit pas y avoir d'obstruction non plus entre la caméra et la surface de l'objet. La distance entre les deux doit respecter une autre plage d'opération de 10 à 20 centimètres. La vitesse de balayage doit être inférieure à 10 centimètres par seconde.

L'ordinateur exécute le logiciel du MapScan qui est en mode d'acquisition sur la photographie à la figure 4.1. L'écran affiche la fenêtre du logiciel où les surfaces partielles de l'objet acquises antérieurement sont visibles. Les surfaces partielles ont toute une couleur distincte afin de les identifier facilement. Lorsque l'acquisition est terminée, l'algorithme de fusion peut s'exécuter sur l'ensemble des surfaces. L'objet numérisé est une statue de la tête de Sophocle. La figure 4.2 présente une photographie du visage de cette statue.



Figure 4.2 Photographie de la statue de Sophocle

Quelques résultats présentés dans ce chapitre proviennent de la surface de l'objet présenté à la figure 4.2.

Il y a deux façons d'utiliser le programme informatique réalisé pour fusionner des surfaces. La première est un programme exécutable qui se nomme « HSSurfaceMerger.exe » et qui a été généré à partir des classes développées pour fusionner des surfaces acquises antérieurement. Il suffit de fournir deux paramètres au programme, soient le nom du fichier qui contient les surfaces à fusionner et le nom d'un fichier dans lequel la surface résultante sera contenue. Ces fichiers doivent nécessairement porter l'extension « i3d ».

La seconde approche est d'utiliser le logiciel complet du MapScan dont l'interface est présentée à la figure 1.7 ainsi qu'à l'écran de l'ordinateur sur la figure 4.1. À l'aide de ce logiciel, il est possible de numériser de nouvelles surfaces ou encore de charger en mémoire des surfaces acquises antérieurement à partir d'un fichier « i3d ». Dans les deux cas les surfaces sont affichées dans la fenêtre graphique de l'interface du logiciel. En appuyant sur le bouton « Merge Records » de l'interface, le processus de fusion est lancé sur les surfaces affichées. Lorsque le processus est terminé, la surface résultante est affichée et il est possible de la sauvegarder dans un fichier « i3d ». Un utilitaire permet d'effectuer la conversion des fichiers « i3d » au format STL pour rendre les surfaces compatibles avec la plupart des logiciels utilisés en CAO.

4.2 Fusion de surfaces planes

Les résultats présentés dans cette section proviennent d'une surface plane. Étant donné que cette surface n'est pas complexe, l'analyse des résultats obtenus est relativement simple. La figure 4.3 présente les résultats de fusion sur un plan. Toutes les surfaces sont affichées en mode fil de fer.

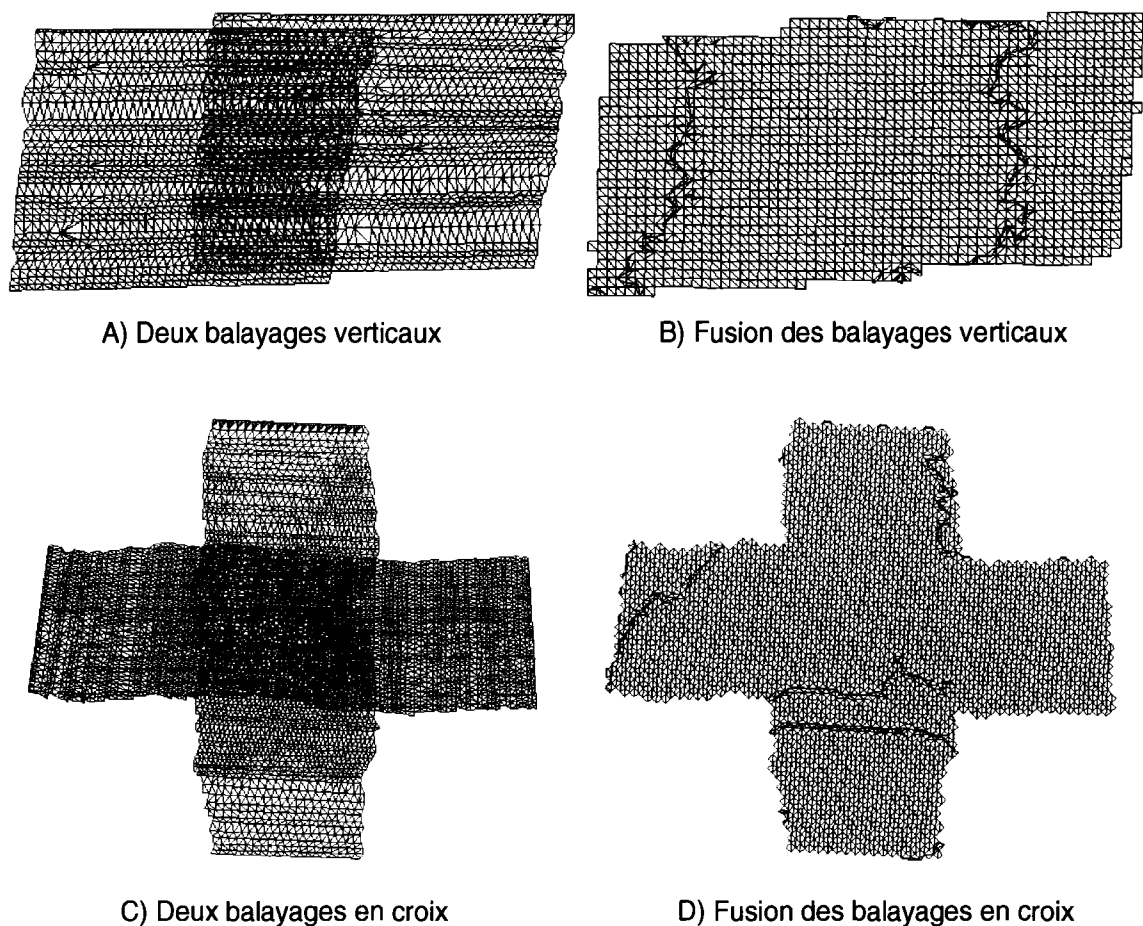


Figure 4.3 Résultats de fusion sur un plan

À la figure 4.3A le plan a été numérisé en deux balayages verticaux selon le point de vue de cette figure. Le balayage de gauche et le balayage de droite se superposent au centre créant une zone de redondance entre les surfaces. Le maillage initial des surfaces à partir du nuage de points s'effectue toujours à partir des points d'un profil vers les points du profil suivant. Ce principe est visible sur cette figure où les triangles forment des rangées pratiquement horizontales entre deux profils voisins. À la figure 4.3B les deux balayages ont été fusionnés et la zone de redondance a été éliminée. À la figure 4.3C le plan a été numérisé en deux balayages en forme de croix. Le premier balayage a été effectué verticalement et le second horizontalement au centre du premier créant une

zone de redondance au centre de la croix. À la figure 4.3D la surface numérisée du plan a été fusionnée et la zone de redondance a été supprimée.

Dans les deux cas (figure 4.3B et 4.3D) les résultats fusionnés représentent bien l'ensemble de la surface numérisée par les deux balayages. La dimension moyenne des triangles est conservée par le processus de fusion. Avant le processus le périmètre moyen des triangles sur les surfaces à la figure 4.3A est de 5,2 millimètres et après le processus sur la surface à la figure 4.3B il est de 5,4 millimètres. Le tableau II à la section 4.7 regroupe des informations numériques comme la dimension moyenne des triangles sur les résultats de la figure 4.3 ainsi que sur ceux des figures suivantes. La configuration des triangles qui forme des lignes sur les surfaces résultantes sera discutée ultérieurement car l'origine de ce phénomène est plus évidente sur certains des résultats suivants (figures 4.5B, 4.8B et 4.10B). Avant le processus de fusion, il y a deux surfaces distinctes et suite à ce processus le modèle final contient une seule surface fusionnée. Pour obtenir la surface fusionnée à partir des deux surfaces individuelles, les fonctions de champ des deux surfaces ont été calculées individuellement. Ensuite ces deux fonctions de champ ont été intégrées en une seule et finalement celle-ci a été triangulée.

4.3 Fusion sans redondance

Un essai a été réalisé en numérisant une surface sans qu'il y ait de redondance entre les balayages. La surface numérisée est une partie d'un morceau de polystyrène qui servait d'emballage protecteur à du matériel informatique. La figure 4.4 présente le résultat de fusion sur ce morceau de polystyrène. Les surfaces sont affichées en mode solide.

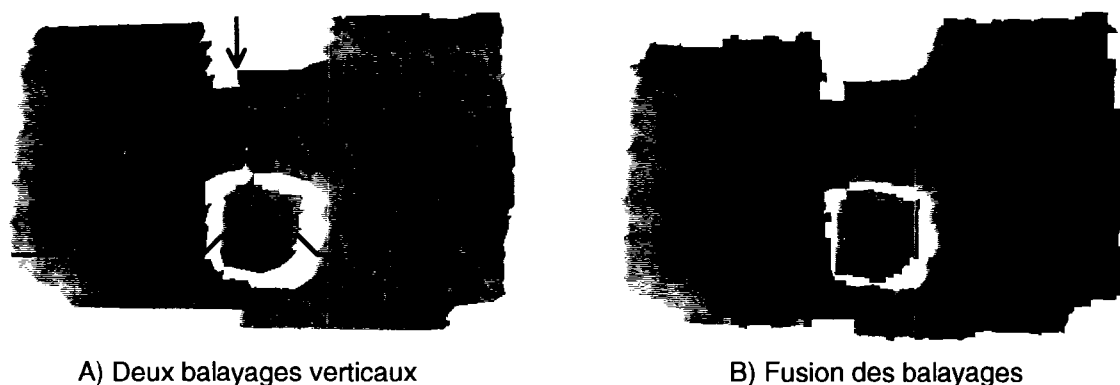


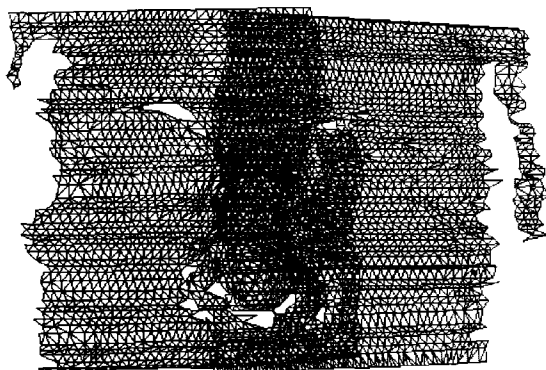
Figure 4.4 Résultat de fusion sur un morceau de polystyrène

À la figure 4.4A, la surface a été numérisée à l'aide de deux balayages verticaux. Dans cet exemple il n'y a pas de redondance entre les deux surfaces, elles sont seulement juxtaposées à l'endroit indiqué par la flèche verte. De plus, le morceau de polystyrène possède une cavité de forme carrée au centre de la surface numérisée. Seulement le fond de cette cavité d'environ 15 millimètres de profondeur a été numérisé étant donné que les parois étaient pratiquement parallèles à l'angle de visée de la caméra. La surface du fond de la cavité semble être dans le même plan que le reste de la surface selon le point de vue à la figure 4.4 mais en réalité elle est 15 millimètres plus profonde. L'espace blanc autour du fond de la cavité représente un manque de données sur les parois de la cavité étant donné que le fond de la cavité est trop loin du reste de la surface pour y être relié par des triangles. La forme du profil laser lors de la numérisation est illustrée sur la surface à la figure 4.4A pour favoriser la perception 3D de cette cavité. Les deux couleurs utilisées sont simplement pour distinguer la limite du champ de vue de la caméra lors des deux balayages. La pente du profil laser dans la troisième dimension est une illustration pour schématiser la profondeur de la cavité. En réalité cette pente est beaucoup plus accentuée et elle tend vers l'infini puisque les quatre parois de la cavité carrée sont perpendiculaires aux surfaces numérisées. Le résultat à la figure 4.4B contient une seule surface fusionnée. Malgré le fait que les surfaces n'avaient aucune région de redondance, l'algorithme d'intégration des fonctions de champ a bien

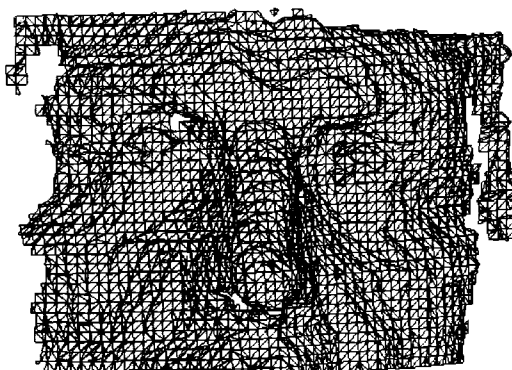
fonctionné étant donné que les contours des deux surfaces étaient à l'intérieur de la distance de redondance.

4.4 Analyse de surfaces typiques

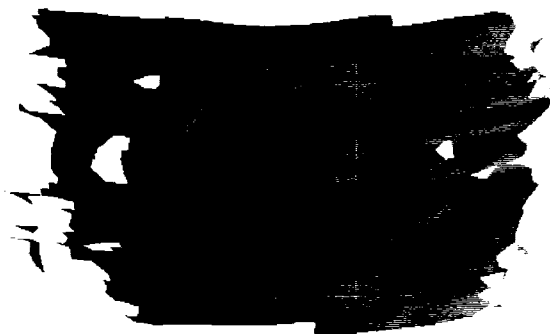
Cette section présente des résultats de fusion sur des portions de la statue de Sophocle. Une analyse des surfaces est réalisée pour décrire quelques imperfections sur les résultats obtenus. Certaines surfaces peuvent contenir des trous ou encore des triangles isolés du reste de la surface. Un phénomène qui s'apparente à des courbes de niveaux est aussi observé sur les surfaces. La figure 4.5 présente les résultats de fusion sur des portions de la statue de Sophocle. Les figures 4.5A et 4.5B sont affichées en mode fil de fer alors que les figures 4.5C et 4.5D le sont en mode solide avec lissage des triangles.



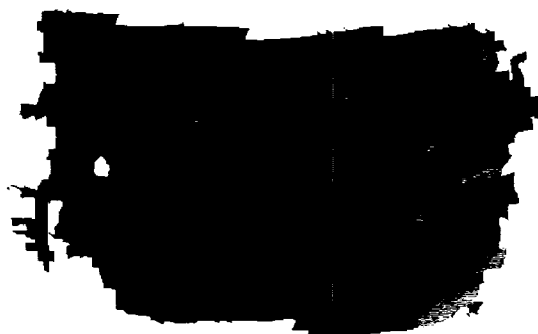
A) Deux balayages verticaux



B) Fusion des balayages



C) Deux balayages verticaux



D) Fusion des balayages

Figure 4.5 Résultats de fusion sur des portions de la statue de Sophocle

La figure 4.5A représente la partie supérieure du visage de Sophocle incluant son front, ses yeux et son nez qui a été numérisée à l'aide de deux balayages verticaux. La zone de redondance est visible au centre. La figure 4.5B présente le résultat de fusion de la figure 4.5A. La grandeur des triangles est toujours conservée dans le résultat. La figure 4.5C représente la partie inférieure du visage de Sophocle incluant sa moustache, sa bouche et son menton qui a également été numérisée à l'aide de deux balayages verticaux. La limite des balayages ainsi que la zone de redondance sont indiquées par les flèches en rouge. La figure 4.5D présente le résultat de fusion de la figure 4.5C.

4.4.1 Trous dans les surfaces

Les deux surfaces de départ à la figure 4.5A contiennent des trous sous les arcades des yeux et sous le nez. Ces trous n'ont pas été triangulés à l'étape du maillage de la surface à partir du nuage de points à cause du manque de données. Dans le cas où les points sont trop éloignés, ils ne sont pas reliés entre eux laissant ainsi un vide. Ceci est préférable à la liaison de ces points qui générerait des trop gros triangles. Le manque de données à un endroit est principalement causé par une vitesse de balayage trop rapide ou encore, comme dans le cas présent, à une partie de la surface ayant une pente très abrupte et pratiquement parallèle à l'angle de vue de la caméra. C'est le même principe que les parois de la cavité à la figure 4.4.

Lorsque ce phénomène est constaté, il suffit d'effectuer de nouveaux balayages locaux aux endroits des trous avec un angle de caméra différent pour obtenir la numérisation complète de la surface d'intérêt. Cette étape n'a pas été réalisée à l'exemple de la figure 4.5A. Cependant dans le résultat à la figure 4.5B les plus petits trous ne sont plus présents et les plus grands sont devenus plus petits. Cette constatation pourrait être vue comme une option intéressante du processus de fusion dans le cas présent mais en réalité ce n'est pas vraiment souhaité car si une pièce possède une ouverture réelle, cette dernière sera diminuée ou bien complètement effacée par le processus de fusion. La bonne méthode pour couvrir les trous causés par le manque de données est d'effectuer

de nouveaux balayages tel que mentionné précédemment. Le même effet de diminution des trous ou des espaces vides est observé aux coins supérieurs droits des figures 4.5A et 4.5B entre les surfaces principales et les franges qui partent de ces coins et qui descendent jusqu'au milieu de la hauteur des surfaces.

Ce phénomène est causé par la résolution de la grille des fonctions de champ. Pour un même trou dans une surface, une grille très précise conservera le trou intact dans le résultat alors qu'une grille moins précise le diminuera et une grille très grossière l'effacera complètement. Donc le fait de prendre une grille plus précise pourrait corriger la situation mais la résolution est choisie automatiquement de façon à conserver la dimension des triangles dans le résultat. Une intervention pourrait être faite à différents niveaux pour éliminer le problème. Le maillage initial pourrait être réalisé de façon plus serrée en modifiant les critères de génération des triangles. Le processus de fusion demeurerait inchangé. Une autre intervention possible serait de modifier le critère de sélection de la résolution de la grille. Le modèle final contiendrait alors beaucoup plus de triangles que les surfaces initiales avant la fusion. Le même phénomène est aussi observé aux figures 4.5C et 4.5D. Les petits trous des surfaces initiales n'existent plus sur la surface résultante et le plus grand trou est devenu plus petit.

4.4.2 Triangles isolés

La surface résultante du processus de fusion n'est pas toujours parfaite. Il arrive que des triangles soient isolés du reste de la surface comme c'est le cas à la figure 4.5B où il y a un petit îlot de triangles isolés en haut, au centre de la surface. Ce phénomène est différent de celui discuté à la section 3.2.3.2 et il se produit à l'occasion lorsqu'il y a une distance significative entre deux parties de deux surfaces qui devraient normalement être parfaitement superposées dans une zone de redondance. Le désalignement entre les deux parties des surfaces est causé par le problème d'étalonnage discuté à la section 2.2.3. Pour que ce phénomène se produise, le désalignement doit être non constant. Il doit être pratiquement nul autour de la région problématique et

assez grand mais toutefois plus petit que la distance de redondance actuelle dans cette région. La surface fusionnée dans cette région passera par le centre des deux parties des surfaces désalignées laissant ainsi quelques triangles isolés à cet endroit. La figure 4.6 illustre le principe des triangles isolés sur la surface.

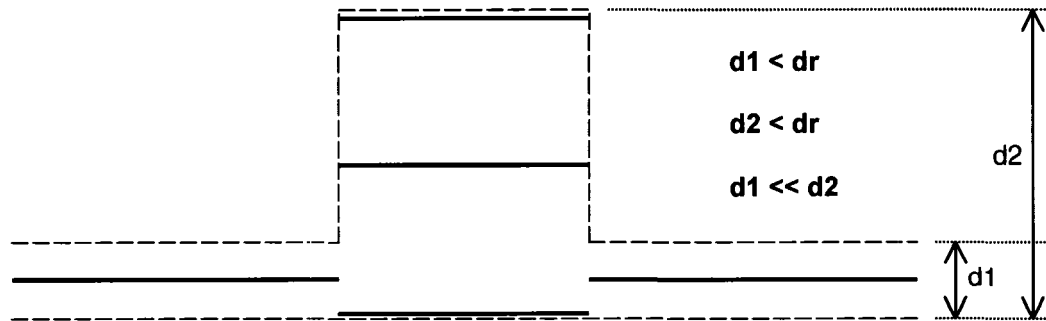


Figure 4.6 Principe des triangles isolés sur la surface

À la figure 4.6, les traits pointillés en bleu et en rouge représentent deux surfaces distinctes à fusionner ensemble. Les segments de droites en noir représentent la surface fusionnée. La distance dr représente la distance de redondance. La surface fusionnée correspond à la moyenne des deux autres surfaces. Le segment central de la surface fusionnée est isolé des autres puisqu'il est beaucoup trop loin de ceux-ci pour y être relié. Ce résultat est une conséquence du grand désalignement entre les surfaces au centre de la figure 4.6. Si la distance de redondance est diminuée de façon à ce qu'elle soit plus petite que la distance $d2$ pour éviter la fusion dans la région problématique, alors le résultat sera similaire. La partie centrale de la figure 4.6 sera considérée comme deux surfaces distinctes et la surface résultante dans cette région sera plutôt représentée par les deux segments verts. Le segment vert supérieur est isolé du reste de la surface car il est beaucoup trop loin de celle-ci pour y être relié alors que celui inférieur peut être relié ou non au reste de la surface en fonction de sa distance ($d1/2$) à celle-ci.

Un algorithme pourrait être élaboré et appliqué sur la surface résultante pour détecter et supprimer les îlots de triangles afin d'obtenir une surface parfaite. Lorsque l'étalonnage du système de positionnement sera terminé, ce phénomène ne devrait plus se produire. Ce phénomène n'est pas présent sur le résultat à la figure 4.5D.

4.4.3 Courbes de niveaux

À la figure 4.5B, la surface résultante contient des régions homogènes séparées par des lignes qui ressemblent à des courbes de niveaux. Ces lignes sont formées par une géométrie locale des triangles qui est différente de celle des triangles voisins. Ce phénomène n'affecte pas la qualité de la surface résultante puisque la déformation est locale et à l'intérieur d'un même voxel. En fait, ces lignes s'apparentent beaucoup à des courbes de niveaux et d'un point de vue 2D, comme sur la figure 4.5B, elles favorisent une perception 3D de la surface qui est plus évidente que celle sur la figure 4.5A. C'est l'algorithme du « marching cube » qui génère cette configuration de triangles qui forment des courbes de niveaux. Une courbe de niveau est produite à chaque fois que la configuration d'un voxel change par rapport à ses voisins. La figure 4.7 présente ce phénomène avec un exemple en 2D. Le principe se généralise en 3D.

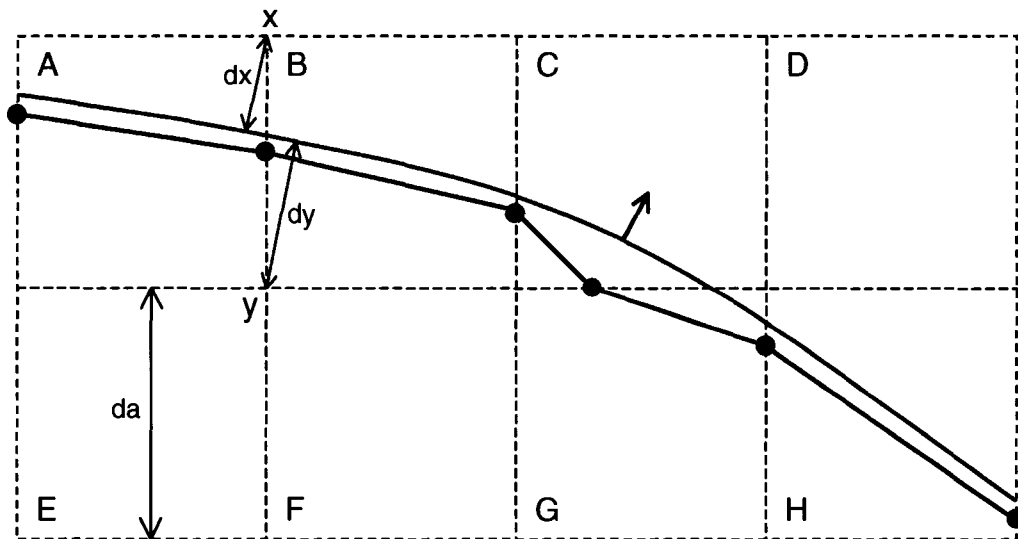


Figure 4.7 Phénomène des courbes de niveaux

À la figure 4.7, la courbe en bleu représente la surface réelle et les segments de droite en vert représentent la surface triangulée. La flèche rouge indique l'orientation de la surface au moyen de son vecteur normal. Les carrés A à H symbolisent les voxels formés avec les éléments de la grille de la fonction de champ. Les points noirs et le point rouge sont les points interpolés sur les arêtes qui forment les sommets des triangles. Il y aura pratiquement toujours une petite erreur entre la surface réelle et la surface triangulée à moins d'avoir une résolution infinie sur la grille. Ceci vient du fait que la somme de dx et dy ne sera pratiquement jamais égale à la longueur d'une arête da étant donné que les distances minimales à la surface dx et dy aux sommets x et y ne sont pas nécessairement à partir du même point sur la courbe qui est sur l'arête entre les sommets x et y . Les distances minimales sont plutôt déterminées à partir des points dont la tangente à la courbe est perpendiculaire au segment de droite qui relie ce point et le sommet d'intérêt.

L'erreur entre les surfaces est presque constante pour des voxels voisins de même configuration. Lorsque la configuration change entre deux voxels voisins, l'erreur

devient significativement plus grande ou plus petite entre la surface réelle et le point interpolé, ce qui produit une irrégularité dans la surface triangulée. Ce phénomène est relativement continu en 3D ce qui donne l'aspect d'une courbe de niveau sur la surface. Les voxels *A*, *B* et *H* à la figure 4.7 présentent une configuration identique avec deux points interpolés sur les arêtes verticales alors que les voxels *C* et *G* présentent une configuration différente des autres avec deux points interpolés, un sur une arête verticale et l'autre sur une arête horizontale. L'erreur au point interpolé en rouge à partir des voxels *C* et *G* est plus grande que les autres erreurs. Dans cet exemple l'erreur au point rouge a été exagérée pour illustrer le principe du phénomène. Plusieurs essais ont été effectués sur des surfaces de synthèse parfaites qui sont définies mathématiquement et l'exemple à la figure 4.7 est tiré d'un essai effectué sur une sphère. L'erreur au point rouge est d'environ 5% alors que les autres erreurs illustrées sont inférieures à 1%. Cette différence est suffisante pour observer le phénomène des courbes de niveaux.

Pour une même surface, une résolution de grille moins précise présente des courbes de niveaux moins fréquentes et plus distancées. Par analogie avec une carte topographique, ceci signifie que l'équidistance entre les courbes est plus grande. Une résolution de grille plus précise présente des courbes plus fréquentes et moins distancées qui ont une équidistance plus petite. En prenant une résolution très précise où la distance entre les éléments de la grille tend vers zéro, le phénomène des courbes de niveaux n'est plus visible sur les surfaces. L'effet des courbes de niveaux est aussi présent sur le résultat à la figure 4.5D mais les courbes ne sont pas visibles en mode solide, elles le sont seulement en mode fil de fer.

4.5 Résultat global

Cette section présente un résultat plus global sur l'ensemble du visage de la statue de Sophocle. La figure 4.8 présente l'ensemble des surfaces partielles ainsi que le résultat de fusion du visage complet en mode fil de fer.

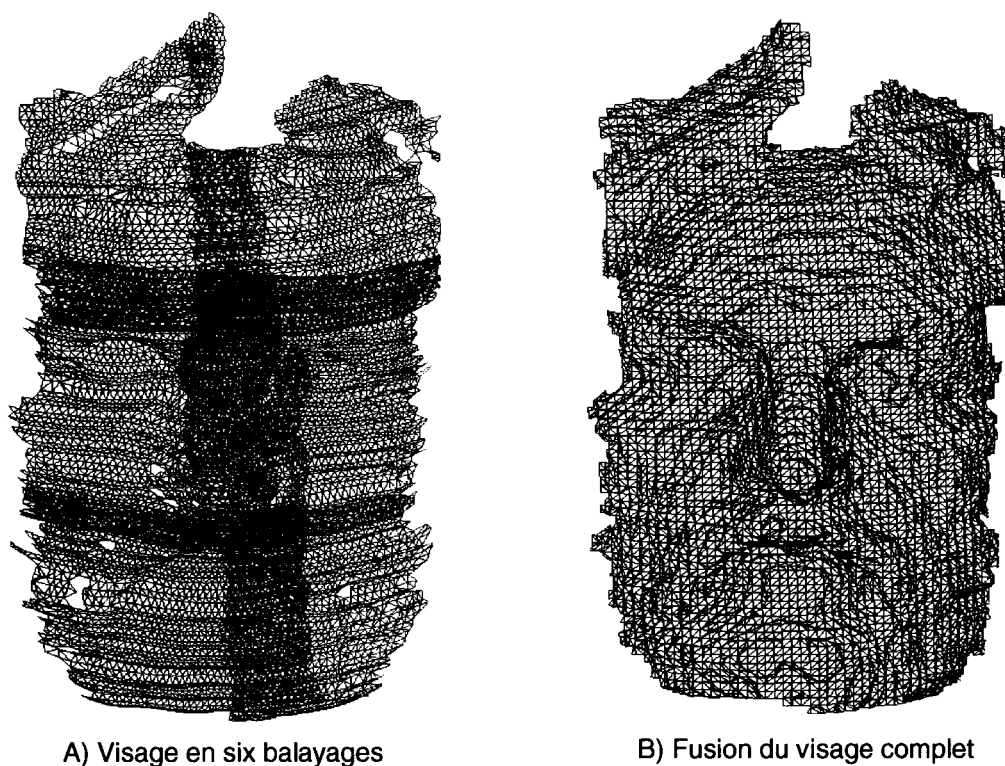


Figure 4.8 Résultat de fusion du visage complet de Sophocle

À la figure 4.8A, le visage de la statue de Sophocle a été numérisé en six balayages, soient deux en largeur et trois en hauteur. Les zones redondantes sont visibles aux régions communes entre les surfaces. La figure 4.8B présente le résultat de fusion de ces six surfaces individuelles. Le résultat couvre l'ensemble du visage de la statue et il contient une seule surface sans redondance. Les courbes de niveaux sont présentes sur le résultat qui reflète bien l'objet numérisé. La figure 4.9 présente les mêmes surfaces selon un point de vue de profil et plus rapproché que celui à la figure 4.8.

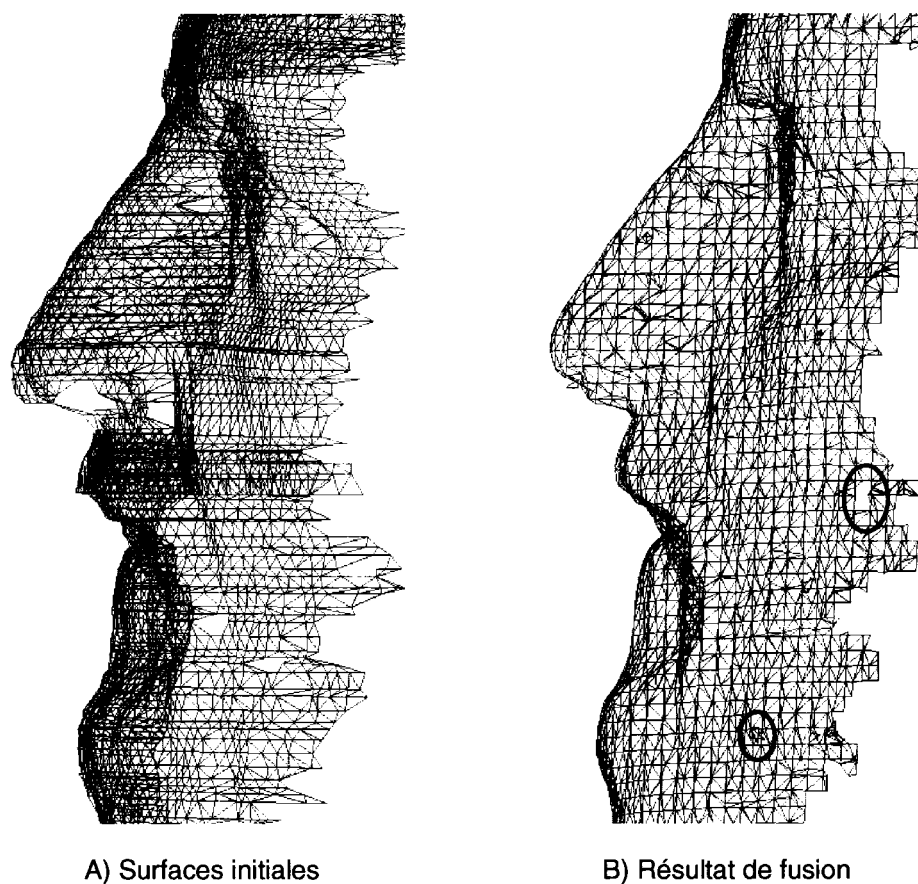


Figure 4.9 Résultat de fusion du visage de Sophocle vue de profil

Quelques petits trous et triangles isolés sont présents sur le résultat de fusion mais ces imperfections ne sont toutefois pas visibles à la figure 4.8B puisque le point de vue est trop loin de la surface. Sur la surface de la figure 4.9B, un trou ainsi qu'un triangle isolé sont encadrés en rouge. Certains détails du visage comme les cheveux et la barbe ne peuvent pas être représentés sur les surfaces initiales ainsi que sur la surface résultante étant donné la précision actuelle du système de numérisation. Les détails de la barbe visibles sur la photographie de la figure 4.2 sont remplacés par une surface plutôt lisse à la figure 4.9B.

4.6 Erreur induite par l'algorithme

L'algorithme de fusion induit une erreur sur les surfaces. La surface d'un moulage du dessous d'un pied a été numérisée et fusionnée afin de calculer l'erreur induite par l'algorithme sur cette surface. La figure 4.10 présente le résultat de fusion sur le moulage d'un pied. Les surfaces sont affichées en mode fil de fer.

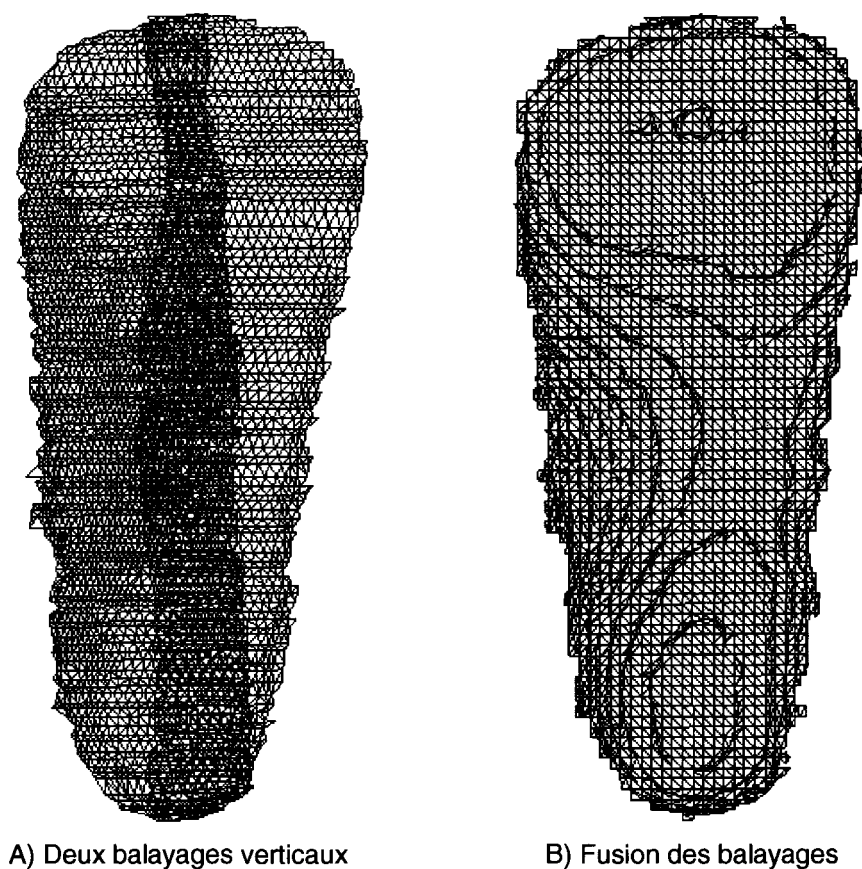


Figure 4.10 Résultat de fusion sur le moulage d'un pied

La figure 4.10A contient deux balayages qui ont été effectués verticalement. La zone de redondance est visible au centre. Le principe du maillage initial des surfaces d'un profil au suivant est aussi visible sur cette figure étant donné que l'objet est relativement plat.

La figure 4.10B contient une seule surface fusionnée sans redondance. L'effet des courbes de niveaux est très évident sur celle-ci.

La surface à la figure 4.10B a été utilisée pour mesurer l'erreur induite sur la surface réelle par le processus de fusion. Le même moulage du pied utilisé dans cet exemple a été numérisé à l'aide d'une caméra 3D à balayage laser (modèle Hyscan C45 de la compagnie Hymarc) montée sur une machine à mesurer les coordonnées (modèle Bright UC200 de Mitutoyo) et qui possède une précision de l'ordre de 25 micromètres. Cette précision est de beaucoup supérieure à celle du MapScan. La surface résultante de la numérisation avec la caméra Hyscan peut donc être considérée comme la surface de référence. La surface de la figure 4.10B a été comparée à la surface de référence à l'aide d'un logiciel de traitement et d'inspection 3D (PolyWorks de InnovMetric). Le module IMAlign de ce logiciel a été utilisé pour mettre en correspondance les deux surfaces, celle de référence avec le résultat de la numérisation à l'aide du MapScan. Ensuite le module d'inspection IMInspect a été utilisé pour déterminer l'erreur entre les surfaces. Le tableau I présente les résultats de la comparaison entre les surfaces générés par le module IMInspect.

Tableau I
Erreur induite par le processus de fusion

Comparison Type	Data to Reference (Data Point)	Data to Reference (Data Point)
Compared Object	moule_pied_1record.stl	moule_pied_2record.stl
Reference	moule_pied_cmm.pol	moule_pied_cmm.pol
Maximum Distance (mm)	4,000000	4,000000
Error Direction	Shortest Distance	Shortest Distance
# Compared Points	1482	3194
Mean (mm)	-0,105842	-0,158837
Standard Deviation (mm)	0,692820	0,742537
Max Error (Positive) (mm)	3,147607	3,098554
Max Error (Negative) (mm)	-3,839385	-3,948089
Points within +/- (1 * StdDev)	1194 (80,566802%)	2588 (81,026925%)
Points within +/- (2 * StdDev)	1403 (94,669366%)	2999 (93,894803%)
Points within +/- (3 * StdDev)	1450 (97,840756%)	3119 (97,651847%)
Points within +/- (4 * StdDev)	1464 (98,785425%)	3168 (99,185974%)
Points within +/- (5 * StdDev)	1478 (99,730094%)	3191 (99,906074%)
Points within +/- (6 * StdDev)	1482 (100,000000%)	3194 (100,000000%)
Tolerance (High) (mm)	2,000000	2,000000
Tolerance (Low) (mm)	-2,000000	-2,000000
# Points Out Of Tolerance (High)	39 (2,631579%)	120 (3,757044%)
# Points Out Of Tolerance (Low)	141 (9,514170%)	359 (11,239825%)

La troisième colonne du tableau I contient les résultats de la comparaison entre la surface à la figure 4.10B et celle de référence obtenue avec la caméra Hyscan. La deuxième colonne contient les résultats de la comparaison entre la même surface de référence et une autre surface. L'autre surface est la surface partielle de gauche seulement de la figure 4.10A. La figure 4.10A contient deux surfaces partielles qui ont été traitées par le processus de fusion pour produire la surface de la figure 4.10B. Seulement la surface de gauche de la figure 4.10A a été isolée et elle a été traitée par le processus de fusion. Le résultat de cette opération est une surface partielle du moulage du pied qui correspond à la même région que la surface de gauche seulement de la figure 4.10A. Étant donné qu'une seule surface a été présentée en entrée au processus de fusion, il n'y a pas eu de réelle fusion de surface. La fonction de champ de l'unique surface a été calculée et ensuite cette même fonction de champ a été triangulée pour

produire une nouvelle surface polygonale. C'est ce résultat qui est comparé à la surface de référence dans la deuxième colonne du tableau I.

La surface de référence couvre l'ensemble de la surface du moulage du pied comme la surface à la figure 4.10B. Le module d'inspection IMInspect permet de comparer deux surfaces qui ne couvrent pas nécessairement toute la même région. Dans le cas présent, le module d'inspection utilise seulement la portion nécessaire de la surface de référence qui correspond à la même région que l'autre surface à comparer. Comme il n'y a pas eu de réelle fusion et simplement une nouvelle triangulation de la surface, cette comparaison permet d'évaluer l'erreur induite par l'algorithme de triangulation « marching cube » seulement. La comparaison avec la surface à la figure 4.10B permet d'évaluer l'erreur induite par l'ensemble du processus de fusion incluant l'intégration des surfaces dans le domaine implicite ainsi que la triangulation finale. En connaissant l'erreur globale et celle de l'algorithme de triangulation il est possible d'évaluer la partie de l'erreur attribuable à l'intégration des surfaces. La deuxième colonne du tableau I présente ainsi l'erreur induite par la triangulation « marching cube » seulement alors que la troisième colonne présente l'erreur combinée de la fusion et de la triangulation.

La méthode utilisée pour déterminer l'erreur est le calcul de la distance minimale entre chacun des points de la surface à comparer et le point le plus près sur la surface de référence. Un seuil de 4 millimètres a été imposé pour limiter l'erreur. Tous les points dont la distance minimale est supérieure à ce seuil ne sont pas considérés dans le calcul de l'erreur dans le but d'éliminer les points erronés qui sont loin des autres et de la surface. Le nombre de points considérés et non considérés sont indiqués au tableau I. Lorsque le calcul est terminé pour tous les points, la moyenne et l'écart type sont calculés. Ces valeurs ainsi que les erreurs positives et négatives maximales rencontrées sont indiquées au tableau I. Pour illustrer l'allure de la distribution de l'erreur, le

nombre de points à l'intérieur des multiples de l'écart type est également indiqué au tableau I.

Selon les écarts types, les erreurs sont relativement faibles. Elles sont du même ordre de grandeur que la précision du système de numérisation. La majeure partie de l'erreur induite provient de la triangulation alors que la fusion introduit seulement une légère erreur supplémentaire à celle de la triangulation pour former l'erreur induite globale. Selon la moyenne, les erreurs maximales et le nombre de points à l'extérieur des tolérances, les surfaces n'étaient pas parfaitement en correspondance avec la référence. L'erreur réellement induite par le processus de fusion est probablement un peu inférieure à celle obtenue. Dans les deux cas comparés, environ 81% des points sont à l'intérieur d'un écart type. Ceci signifie que la majeure partie des erreurs est concentrée dans cette plage. La dispersion des erreurs est plus faible que celle d'une distribution normale où environ 67% des données se situent à l'intérieur d'un écart type. Comme pour une distribution normale, la presque totalité des erreurs est contenue à l'intérieur de trois écarts types.

4.7 Performances de l'algorithme

Les performances de l'algorithme de fusion ont été mesurées pendant et après son exécution sur les surfaces des résultats présentés dans ce chapitre. Les tableaux II et III contiennent plusieurs informations numériques concernant l'algorithme de fusion. Le tableau II contient des informations sur les surfaces initiales et résultantes du processus telles que le nombre de triangles et le périmètre moyen de ceux-ci avec les différences entre les surfaces initiales et résultantes. Le tableau III contient des informations sur le processus de fusion telles que le temps de calcul de l'algorithme, la mémoire maximale utilisée pendant le calcul, la résolution de la grille utilisée pour les fonctions de champ, le nombre d'éléments traités et le temps de calcul moyen par élément.

Tableau II
Informations numériques sur les surfaces

Figure	Nb triangles initiaux	Nb triangles finaux	Delta Nb triangles (%)	Pér. moyen initial (mm)	Pér. moyen final (mm)	Delta pér. moyen (%)
4.3B	4923	3480	29,31	5,217	5,409	3,68
4.3D	8251	6158	25,37	7,868	8,108	3,05
4.4B	6959	5980	14,07	4,409	4,596	4,24
4.5B	7934	7242	8,72	6,349	6,259	1,42
4.5D	5335	5033	5,66	6,307	6,364	0,90
4.8B	18208	15217	16,43	6,472	6,381	1,41
4.10B	8080	6176	23,56	7,364	7,694	4,48

Le nombre de triangles au tableau II est d'environ 5% à 30% plus petit dans le modèle final que dans la somme des surfaces initiales. Ceci est tout à fait normal puisque les zones de redondance sont supprimées dans la surface finale. L'erreur maximale rencontrée sur le périmètre moyen des triangles de la surface finale par rapport à celui de l'ensemble des surfaces initiales est de 4,48%. La dimension moyenne des triangles est donc conservée dans le modèle final. Dans cinq cas sur sept le périmètre moyen des triangles de la surface finale est plus grand que celui des surfaces initiales. L'algorithme a donc tendance à augmenter légèrement la dimension des triangles.

Tableau III
Informations numériques sur l'algorithme de fusion

Figure	Temps de calcul (s)*	Mémoire utilisée (Mo)	Résolution grille (mm)	Nb éléments traités	Temps moyen élément (µs)*
4.3B	12,3	3,720	2,229	30687	400,8
4.3D	16,8	6,128	3,373	41602	403,8
4.4B	24,9	5,716	1,880	48614	512,2
4.5B	27,5	5,216	2,717	42907	640,9
4.5D	16,3	3,752	2,699	30295	538,0
4.8B	65,4	11,100	2,770	102504	638,0
4.10B	17,5	5,064	3,155	39332	444,9

* L'ordinateur utilisé pour effectuer les calculs dans les temps indiqués est un PC monoprocesseur de type Pentium 3 ayant une horloge de 866 MHz avec une fréquence d'accès de 100 MHz à la mémoire vive. Le système d'exploitation utilisé est Windows NT 4.0.

Les temps de calcul au tableau III varient de 12 à 65 secondes pour les différents résultats présentés. L'utilisateur doit attendre un certain temps avant d'obtenir le résultat. Ce délai est toutefois acceptable compte tenu de la complexité et du nombre d'étapes du processus de fusion. Avant la mise en œuvre des techniques d'optimisation, les temps de calcul auraient été de l'ordre de 3,3 à 18,1 heures environ. Le temps de calcul dépend de plusieurs paramètres dont certains ont une relation entre eux. Il dépend du nombre de surfaces, de la forme et du volume des surfaces, du nombre et de la dimension des triangles des surfaces ainsi que de la résolution et du nombre d'éléments utiles de la grille. Il n'est donc pas possible d'associer le temps de calcul à un seul paramètre en particulier dans le tableau III.

Cependant un facteur important est celui du nombre total d'éléments traités qui varie de 30 295 à 102 504. Le nombre d'éléments traités représente le nombre d'éléments total, dans toutes les fonctions de champ des surfaces initiales, qui ont été étiquetés comme étant utiles à leur fonction de champ et pour lesquels des valeurs de distances ont été calculées. Ce facteur est important étant donné que le temps moyen de traitement d'un élément ne varie pas énormément d'un résultat à l'autre. Ces temps sont de 400 à 640 microsecondes selon le cas. Le principal facteur qui influence ce dernier paramètre est le nombre de triangles dans chacun des éléments du plan de projection. Le point le plus près de l'élément de la grille sur la surface doit être évalué pour tous les triangles contenus dans l'élément du plan de projection. Cette évaluation est de loin l'étape la plus longue de toutes les étapes du processus.

La mémoire vive utilisée par le processus est de 3,7 à 11,1 mégaoctets. Ces valeurs représentent la quantité de mémoire maximale utilisée par le processus qui nécessite plus ou moins de mémoire en cours d'exécution. Les valeurs présentées sont le résultat d'une certaine gestion de la mémoire. Au début du processus les surfaces initiales sont chargées en mémoire. Lorsque toutes les fonctions de champ individuelles sont calculées, la mémoire qui contenait les surfaces initiales est libérée. Lorsque la fonction

de champ intégrée est calculée, la mémoire qui contenait toutes les fonctions de champs individuelles est libérée. Lorsque la triangulation est terminée, la mémoire qui contenait la fonction de champ intégrée est libérée. À la fin du processus, la surface résultante est sauvegardée et la mémoire qui la contenait est aussi libérée. Les valeurs maximales sont raisonnables compte tenu de la quantité de mémoire présentement disponible dans les ordinateurs. Ces valeurs étaient environ dix fois plus grandes avant l'élaboration de l'optimisation des éléments utiles.

La résolution de la grille varie de 1,9 à 3,4 millimètres environ. Cette variation n'est pas très grande et elle est directement liée à la variation du périmètre moyen des triangles dans les surfaces initiales. Ceci signifie que pour tous les résultats présentés, les grandeurs moyennes des triangles des surfaces initiales sont toutes semblables. C'est normal puisque dans tous les cas un sous-échantillonnage d'un point sur huit a été utilisé sur les profils et les vitesses de balayage étaient semblables.

4.7.1 Distributions des périmètres

En plus de comparer le périmètre moyen des triangles avant et après le processus de fusion, des histogrammes de distribution des périmètres ont été générés pour vérifier de façon plus détaillée si la dimension des triangles était bien conservée par le processus de fusion. La figure 4.11 présente la distribution des périmètres pour les surfaces du moulage du pied présenté à la figure 4.10.

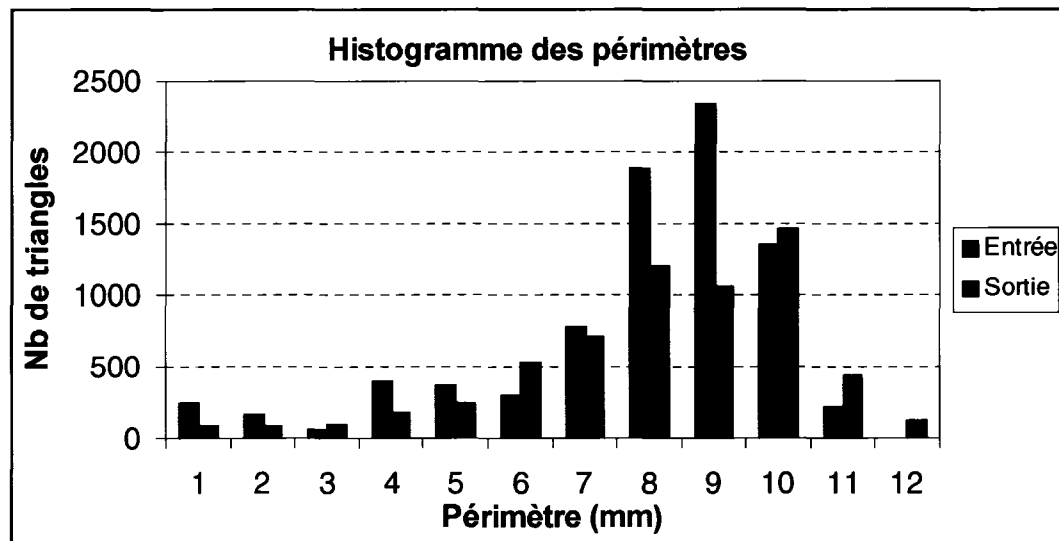


Figure 4.11 Distribution des périmètres pour le moulage du pied

Sur le graphique de la figure 4.11, les barres au-dessus du nombre 1 en abscisse représentent un certain nombre de triangles indiqué en ordonnée qui ont un périmètre entre zéro et un millimètre. Celles au-dessus du nombre 2 représentent des périmètres de un à deux millimètres et ainsi de suite. Les barres en bleu représentent le périmètre des triangles sur l'ensemble des surfaces initiales et celles en vert représentent le périmètre des triangles sur la surface finale. La somme du nombre de triangles total indiqué par les barres bleues est évidemment plus grande que celle des barres vertes puisqu'il y a plus de triangles dans les deux surfaces initiales que dans la surface résultante. Les distributions ne sont pas identiques mais elles sont suffisamment semblables pour se fier sur le critère du périmètre moyen afin d'obtenir des triangles de même dimension sur la surface résultante. Concernant les deux distributions, la plupart des triangles ont des périmètres entre six et dix millimètres. Dans l'ensemble, les histogrammes des autres résultats présentés se comparent à celui de la figure 4.11. La distribution des périmètres des triangles est semblable avant et après le processus de fusion.

CONCLUSION

Le présent projet de maîtrise consistait à développer et à implémenter une méthode de fusion de surfaces provenant d'un profilomètre tenu à la main. Une revue de la littérature a permis d'identifier deux grandes familles de méthodes possibles pour réaliser cette tâche. Il s'agit des méthodes volumétriques et des méthodes surfaciques. Toutes les méthodes de ces familles ont été testées par les chercheurs sur des images de profondeur seulement. Certaines de ces méthodes seraient possiblement adaptables aux données obtenues à l'aide du profilomètre tenu à la main. D'autres méthodes comme celle des modèles solides ou encore celle des surfaces paramétriques sont intéressantes mais elles ne sont pas compatibles avec les données du MapScan. La méthode de fusion de surfaces par la représentation implicite a été retenue pour réaliser ce projet. Cette méthode a été testée sur des données similaires à celles du MapScan. C'est une méthode relativement efficace pour des nuages de points non structurés dans l'espace 3D.

Les surfaces à fusionner en une seule sont représentées dans un autre domaine que celui qui les définit explicitement par un ensemble de polygones. Un algorithme a été développé pour convertir les surfaces dans le nouveau domaine implicite de la fonction de champ. Une méthode automatisée a été élaborée pour calculer la résolution de la grille des fonctions de champ afin de conserver la grandeur des triangles dans le modèle final. Des méthodes pour optimiser le temps de conversion des surfaces ainsi que la mémoire nécessaire pour les supporter dans le domaine implicite ont aussi été élaborées. Des essais préliminaires ont été faits sur les fonctions de champ et ce domaine de représentation des surfaces dans l'espace 3D offre un potentiel intéressant pour la manipulation de celles-ci en vue d'améliorer leur qualité par des opérations de filtrage par exemple.

Un algorithme a été développé pour fusionner l'ensemble des fonctions de champ en une seule intégrée. Cet algorithme est adapté à la structure particulièrement optimisée de la représentation implicite. Les paramètres de cet algorithme ont été ajustés pour optimiser le fonctionnement de la fusion en fonction des performances actuelles du système de numérisation. Deux algorithmes de triangulation ont été développés pour effectuer le maillage polygonal du modèle final à partir de la fonction de champ intégrée. Il s'agit du « marching triangle » et du « marching cube ». Plusieurs améliorations ont été apportées à l'algorithme du « marching triangle » pour le rendre plus efficace et l'adapter aux données du MapScan. Cependant les résultats obtenus à l'aide de l'algorithme amélioré n'étaient pas satisfaisants et celui-ci a été abandonné. Différentes options ont toutefois été présentées en vue d'un développement futur pour améliorer davantage cet algorithme. Le second algorithme, le « marching cube », a aussi été amélioré, optimisé et adapté en fonction du type de données actuellement traitées. Les temps de calcul et la qualité des résultats obtenus à l'aide de cet algorithme sont satisfaisants.

Des essais ont été réalisés sur plusieurs surfaces différentes pour valider l'ensemble du projet. La surface résultante représente bien l'ensemble des surfaces initiales sans les zones de redondances qui ont été éliminées. En général, la dimension moyenne des triangles est conservée sur la surface résultante par rapports aux surfaces initiales. Le processus de fusion a toutefois tendance à diminuer les trous présents dans la surface. La cause du phénomène est connue et des solutions potentielles ont été présentées. Celles-ci auront cependant une influence sur le modèle final. Il arrive que le résultat contienne quelques triangles isolés du reste de la surface. La cause de ce phénomène est aussi connue et celui-ci devrait être corrigé ultérieurement lors de la finalisation du développement du MapScan. L'algorithme de triangulation génère des courbes de niveaux sur la surface résultante. Ceci n'affecte pas la qualité du résultat. Les courbes de niveaux sont plutôt un avantage lorsque les résultats de fusion sont présentés en 2D, sur une feuille de papier par exemple. L'utilisation d'une grille plus précise atténue le

phénomène des courbes de niveaux. Le niveau de détails disponibles au résultat est limité par la précision du système de numérisation. Ceci n'a rien à voir avec le processus de fusion, les détails sont perdus à l'étape de numérisation et ils ne sont pas présents sur les surfaces initiales.

L'erreur induite par le processus de fusion a été quantifiée et elle est du même ordre de grandeur que la précision théorique du système de numérisation. La principale portion de l'erreur provient de l'étape de la triangulation. Un tableau comparatif présente certaines différences mineures et justifiées, apportées par le processus de fusion entre la surface résultante et les surfaces initiales. Deux autres tableaux présentent certains paramètres et les performances du processus de fusion sur les résultats présentés. Les performances du processus sont relativement bonnes sur tout ordinateur qui utilise une technologie récente. Il ne s'exécute cependant pas en temps réel, il faut de quelques secondes à quelques minutes de traitement, selon le nombre de polygones formant les surfaces, pour obtenir le résultat. En plus d'avoir une dimension moyenne des triangles conservée, la distribution des périmètres des triangles est aussi semblable avant et après le processus de fusion.

Le domaine de représentation implicite par la fonction de champ des surfaces dans l'espace 3D est un outil potentiellement intéressant pour effectuer des opérations sur les surfaces lors de développements futurs. Des essais ont démontré qu'il est possible d'effectuer du filtrage en 3D sur les surfaces dans ce domaine. D'autres types d'opérations comme des transformations géométriques sont aussi possibles. Le dimensionnement par facteur d'échelle a été réalisé sur des surfaces à l'aide de la fonction de champ. L'utilisation de cette représentation peut possiblement être utile à d'autres applications comme la compression de modèles, le traitement volumétrique, la corrélation et la mise en correspondance de surfaces.

La version actuelle du MapScan est fonctionnelle et elle répond au besoin de certaines applications. Des développements futurs pour améliorer le MapScan sont envisagés afin qu'il puisse être utile à de nouvelles applications. Le système de positionnement pourrait être remplacé par un autre système plus précis. Le nouveau système pourrait être basé sur une technologie inertielle, optique ou autre. L'utilisation d'un système hybride intégrant plus d'une technologie permettrait possiblement d'accroître davantage sa précision globale.

L'intégration des différentes composantes du MapScan au boîtier du profilomètre augmenterait sa portabilité, son autonomie et la liberté de mouvements lors de son utilisation. Le nouveau système de positionnement pourrait être conçu en fonction de son intégration au boîtier sans aucune composante externe. L'ordinateur utilisé avec le MapScan pourrait aussi être intégré au profilomètre ou encore dans un autre boîtier compact portable à la taille ou en bandoulière. La disponibilité actuelle des ordinateurs sur une seule carte de petites dimensions rend cette intégration possible. Un petit écran comme celui d'un appareil photo numérique ou celui d'un caméscope serait incorporé au boîtier.

La caméra actuelle du profilomètre est utilisée pour acquérir les images du profil laser. Cette même caméra ou une seconde caméra installée près de la première pourrait être utilisée pour acquérir des images standards de la surface de l'objet à numériser. La couleur et la texture de la surface pourraient être extraites des images et appliquées sur le modèle polygonal final pour obtenir un rendu de la surface pratiquement identique à la surface de l'objet.

RECOMMANDATIONS

Afin d'améliorer davantage les performances et les résultats obtenus avec le processus de fusion des surfaces lors de développements futurs, quelques aspects du projet pourraient être modifiés. Si les temps de calcul des algorithmes sont jugés encore trop long, il est possible de les diminuer de différentes façons. Sans rien changer aux algorithmes, l'utilisation d'un ordinateur ayant une puissance de calcul plus élevée améliorera évidemment les temps de calcul. Une autre option serait de faire une version parallèle du projet destinée aux ordinateurs multiprocesseurs. Selon le nombre de processeurs disponibles, ceci diminuerait grandement les temps puisque le traitement peut être parallélisé entièrement et facilement. En effet, les algorithmes se prêtent bien au traitement parallèle.

Le calcul des fonctions de champ des surfaces initiales est indépendant pour chacune d'elles et un fil logiciel par fonction pourrait être lancé. S'il y a encore plus de processeurs disponibles que de surfaces à fusionner alors le calcul d'une seule fonction de champ pourrait aussi être parallélisé facilement puisque le calcul pour un élément est indépendant des autres. Le volume de la grille pourrait être divisé en plusieurs portions et un fil par portion serait lancé ou encore, selon l'approche privilégiée, un nouveau fil pourrait être créé pour chacun des éléments. L'intégration des fonctions de champ en une seule peut aussi être parallélisée au même titre que le calcul d'une fonction de champ puisque l'intégration se fait par élément et indépendamment des autres. Le calcul de l'intégration pourrait même débiter avant la fin des calculs pour évaluer les fonctions de champ en autant que toutes les fonctions de champ sont calculées pour un élément en particulier avant d'effectuer l'intégration pour cet élément. La triangulation de la fonction de champ intégrée pour obtenir le modèle final se parallélise aussi bien que le reste du procédé étant donné qu'elle est effectuée par voxel et indépendamment des autres. Encore une fois, un fil par portion du volume global ou par voxel pourrait

être créé. Dans le même ordre d'idée que précédemment, la triangulation pourrait débiter avant la fin de l'intégration, donc avant la fin de l'évaluation des fonctions de champ aussi, en autant que les valeurs intégrées des huit sommets du voxel en traitement sont calculées.

Comme l'ensemble des algorithmes du procédé ne dépend pas du voisinage, une toute autre approche pourrait être prise pour réaliser ce projet afin de diminuer encore davantage le temps de calcul. Il s'agit d'une méthode inverse à celle implantée en fonction de l'ordre des opérations. Au départ, le volume global de toutes les surfaces pourrait être divisé par une grille 3D en fonction de la résolution désirée. Pour un voxel en particulier de la grille, la triangulation serait appliquée. Pour trianguler, il faut connaître les valeurs de la fonction de champ intégrée aux huit sommets du voxel. Ces valeurs seraient calculées avec l'algorithme d'intégration des fonctions de champ. Pour calculer la valeur intégrée d'un sommet en particulier d'un voxel, il faut évaluer la distance minimale à la surface de tous les éléments pertinents au calcul. Il s'agit de vérifier pour toutes les fonctions de champ si cet élément à ce point de l'espace existe, c'est-à-dire s'il est à l'intérieur du domaine de la fonction de champ. Si c'est le cas, alors il faut vérifier si l'élément est utile à sa fonction. Si l'élément est utile, alors le calcul de la valeur de la fonction de champ est fait pour cet élément. Un nombre maximum d'éléments correspondant au nombre de fonctions de champ serait calculé. À partir des valeurs de fonctions de champ calculées, l'intégration serait calculée pour un sommet en particulier. Le processus serait répété pour calculer les valeurs intégrées aux autres sommets du voxel. Ensuite la triangulation du voxel serait réalisée. L'algorithme passerait au voxel suivant de la grille et tout le processus serait répété pour le nouveau voxel jusqu'au traitement du dernier voxel de la grille.

Le gain en temps de calcul serait réalisé en partie grâce à l'amélioration apportée à l'algorithme de triangulation qui est présentée à la section 3.2.3.2 de ce document. Aussitôt que deux sommets voisins d'un voxel auraient des valeurs intégrées de

distances opposées en signe et plus grande que la résolution de la grille alors le voxel serait rejeté et aucun triangle ne serait généré à l'intérieur de ce voxel. Il ne serait donc pas nécessaire de calculer les valeurs des autres sommets du voxel. L'autre partie du gain serait réalisée grâce à l'optimisation des éléments utiles présentée à la section 2.1.3.1 de ce document. Aussitôt qu'un sommet du voxel serait étiqueté comme étant non utile à la fonction de champ alors le voxel serait aussi rejeté diminuant encore le nombre de sommets à évaluer. Un seul sommet non évalué représente beaucoup de calculs en moins à commencer par un calcul d'intégration et plusieurs calculs de fonctions de champ. Ceux-ci représentent l'étape la plus longue de toutes les étapes du processus. Les valeurs intégrées des sommets évalués pour un voxel seraient conservées pour les voxels voisins puisqu'un voisin immédiat possède quatre sommets communs avec le voxel en traitement. Lorsque tous les voxels communs à un sommet auraient été traités, la valeur intégrée de ce sommet pourrait être effacée de la mémoire puisqu'elle ne serait plus utile.

Cette méthode aurait un autre avantage qui est celui d'utiliser beaucoup moins de mémoire que la méthode implantée car seulement les informations locales au voxel et à ses voisins seraient conservées comparativement à toutes les informations de l'ensemble des voxels du volume global. Cette méthode serait aussi beaucoup plus simple à paralléliser puisqu'il suffirait de créer un fil par voxel. Il faudrait cependant gérer la redondance efficacement afin de calculer un sommet une seule fois. Un sommet déjà calculé ne devrait pas être calculé à nouveau pour un voisin et un sommet présentement en traitement ne devrait pas être traité en parallèle pour un voisin mais le résultat devrait plutôt être attendu du traitement en cours pour ce sommet.

Une autre modification pour diminuer le temps de calcul et la mémoire serait d'utiliser la notion de boîte de contour minimale. Déjà les fonctions de champ sont limitées par les boîtes de contour des surfaces mais ces boîtes sont les boîtes minimales orientées selon les axes du système de coordonnées. Pour chaque surface, il existe une boîte de

contour minimale, orientée de façon quelconque selon la surface, dont le volume est plus petit ou égal à la boîte de contour orientée avec les axes. En calculant et en utilisant ces nouvelles boîtes de contour plutôt que celles utilisées présentement, le nombre d'éléments des fonctions de champ serait plus petit ce qui résulterait en un gain de temps et de mémoire. La différence ne serait cependant pas majeure car le nombre d'éléments utiles serait sensiblement le même et cette modification complexifierait légèrement l'ensemble du code.

Dans le même ordre d'idée, l'optimisation du plan de projection présentée à la section 2.1.3.2 pourrait être modifiée pour choisir le plan de projection optimal. Déjà le meilleur plan de projection parmi les trois plans composés de deux des trois axes du système de coordonnées est sélectionné à l'aide d'un critère de dispersion maximale des triangles dans le plan. Il existe un plan qui offre une dispersion optimale des triangles plus grande ou égale à celle du meilleur plan présentement calculé. Une première approche pour déterminer un plan optimal est de calculer le vecteur normal à ce plan qui est la somme des vecteurs normaux de tous les triangles de la surface. Une seconde approche pour déterminer un plan optimal est d'effectuer une analyse par composantes principales de la matrice de covariance des triangles. Le plan de projection correspondrait au plan parallèle à la surface dominante de l'objet et il serait défini par les deux vecteurs propres dominants de la matrice de covariance. Chacun des éléments du plan optimal contiendrait en moyenne moins de triangles que ceux du plan utilisé actuellement. Ceci résulterait en une recherche moins longue sur moins de triangles pour déterminer le point le plus près d'un élément sur une surface. Le gain ne serait probablement pas énorme étant donné que le nombre moyen de triangles serait seulement un peu plus petit. De plus, ce plan serait orienté de façon quelconque ce qui complexifierait aussi le code d'autant plus que son orientation serait sûrement différente de celle de la boîte de contour minimale.

Pour améliorer les performances de l'algorithme et la qualité de la surface résultante, une grille de résolution variable pour les fonctions de champ pourrait être utilisée. Aux endroits où la surface présente une grande courbure, la résolution pourrait être plus précise afin de mieux la représenter alors qu'aux endroits où elle présente une faible courbure, la résolution pourrait être moins précise afin de limiter le nombre d'éléments. Le fait de supprimer des éléments inutiles représente un gain de temps sur les calculs de fonctions de champ, d'intégration et de triangulation. Cette modification résulterait en une surface finale qui contiendrait des petits triangles pour mieux suivre la surface où elle varie rapidement et des grands triangles pour représenter des régions planes. La taille du modèle final en terme d'espace mémoire serait probablement plus petite puisqu'un grand triangle remplacerait plusieurs petits pour décrire une même région.

Il faudrait élaborer une méthode pour déterminer la résolution en fonction de la géométrie locale de la surface. Cette modification complexifierait le code à l'étape de l'intégration. Dans les zones non redondantes il n'y aurait pas de problème mais dans les zones redondantes les résolutions ne seraient pas les mêmes pour toutes les fonctions de champ. En théorie les surfaces dans une zone redondante qui représentent la même partie d'un objet devraient être identiques mais en pratique, elles seront sensiblement différentes et la résolution aussi. Il faudrait déterminer la résolution dans ces zones pour la fonction de champ intégrée et élaborer une méthode pour aligner et calculer la valeur intégrée.

Pour améliorer la qualité de la surface finale, une modification pourrait être apportée à l'algorithme de triangulation. Cet algorithme a tendance à diminuer la grandeur des trous sur la surface et des solutions ont déjà été proposées pour corriger la situation. Cependant les solutions proposées affecteraient la grandeur et le nombre de triangles de la surface résultante. Présentement l'algorithme utilise seulement la valeur de la distance à la surface de la fonction de champ intégrée pour trianguler la surface. Une méthode pourrait être élaborée afin que l'algorithme utilise aussi l'information de

contour disponible pour que la grandeur des trous soient conservée. De cette façon la résolution de la grille et la grandeur des triangles des surfaces initiales ne seraient pas modifiées et la grandeur et le nombre de triangles de la surface résultante demeureraient pratiquement inchangés. La plus simple des méthodes serait de ne pas trianguler un voxel qui contient des éléments de contour mais ceci aurait peut-être l'effet inverse d'augmenter la grandeur des trous ou d'en créer des nouveaux.

Une autre façon d'améliorer la qualité du résultat serait d'apporter une autre modification à l'algorithme de triangulation. Dans certains cas où la résolution de la grille serait trop grossière, il pourrait arriver que le fait de ne pas trianguler un voxel qui contient un sommet étiqueté comme étant non utile à la fonction de champ, produise un trou dans la surface résultante. Une méthode de triangulation particulière pourrait être élaborée pour arriver à trianguler un voxel qui contient un ou des sommets non utiles pour lesquels aucune valeur intégrée de distance n'a été calculée. Une solution pourrait être d'interpoler des valeurs de distance à ces sommets en fonction des valeurs aux sommets voisins. Une limite devrait possiblement être imposée sur la quantité maximale de sommets non utiles dans un voxel pour que celui-ci soit tout de même triangulé.

ANNEXE 1

Fiche technique du MapScan

Cette annexe contient la fiche technique du MapScan produite à l'INO. Cette fiche est destinée aux clients potentiellement intéressés au produit. La fiche présente le MapScan et ses options ainsi que les étapes pour l'utiliser. Quelques applications pour le MapScan sont suggérées et ses spécifications techniques sont énumérées.

ANNEXE 2

Principaux articles utilisés

Cette annexe contient les trois principaux articles utilisés pour réaliser le projet. Le premier article s'intitule « Implicit surface-based geometric fusion » (Hilton & al, 1998) et il décrit le principe de la fonction de champ ainsi que l'intégration des fonctions en une seule. Le second article s'intitule « Marching triangles: Range image fusion for complex object modeling » (Hilton & al, 1996a) et il décrit l'algorithme de triangulation « marching triangle ». Le troisième article s'intitule « Marching cubes: A high resolution 3D surface reconstruction algorithm » (Lorensen & Cline, 1987) et il décrit l'algorithme de triangulation « marching cube ».

RÉFÉRENCES BIBLIOGRAPHIQUES

Algorri, M-E., Schmitt, F. (1996). « Surface reconstruction from unstructured data », *Computer Graphics*, Vol. 15, No. 1, pp. 47-60.

Bajaj, C., Bernardini, F., Xu, G. (1995). « Automatic reconstruction of surfaces and scalar fields from 3D scans », *Computer Graphics*, Siggraph 95, Los Angeles, USA, August, pp. 109-118.

Besl, P. J., McKay, N. D. (1992). « A method of registration of 3D shapes », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239-256.

Curless, B., Levoy, M. (1996). « A volumetric method for building complex models from range images », *Computer Graphics*, Siggraph 96, New Orleans, USA, August, pp. 221-227.

Edelsbrunner, H., Facello, M.A., Fu, P., Qian, J., Nekhayev, D.V. (1998). « Wrapping 3D scanning data », *Proc. SPIE*, March, Vol. 3313.

Foley, J.D., Van Dam, A., Feiner, S.K., Hughes, J.F. (1997). « Computer graphics: Principles and practice », *Addison-Wesley Publishing Company*, Boston, USA.

Harvey, E., Arsenault, M., Bélanger, B. (1999). « Compact and portable laser system for 3D acquisition: Space applications », *Tech. Rep., Canadian Space Agency*, #CSA-ST-CR-1999-0045.

Harvey, E., Arsenault, M., Lavoie, J.-F., Bélanger, B., Boucher, M.-A. (2001). « Compact and portable 3D camera for space applications », *Third International Conference on 3D Digital Imaging and Modeling*, Quebec, Canada, May, pp. 31-37.

Hilton, A., Illingworth, J. (1997). « Marching triangles: Delaunay implicit surface triangulation », *Tech. Rep., Univ. of Surrey, Dept. of Electronic Eng.* <http://www.ee.surrey.ac.uk/Research/VSSP/3DVision/mt.html>.

Hilton, A., Stoddart, A. J., Illingworth, J., Windeatt, T. (1998). « Implicit surface-based geometric fusion », *Computer Vision and Image Understanding*, Vol. 69, No. 3, pp. 273-291.

Hilton, A., Stoddart, A. J., Illingworth, J., Windeatt, T. (1996a). « Marching triangles: Range image fusion for complex object modeling », *IEEE International Conference on Image Processing*, Lausanne, Switzerland, September, pp. 381-384.

Hilton, A., Stoddart, A. J., Illingworth, J., Windeatt, T. (1996b). « Reconstruction of 3D Delaunay surface models of complex objects », *IEEE International Conference on Systems Man and Cybernetics*, Beijing, China, October, pp. 2445-2450.

Hilton, A., Stoddart, A., Illingworth, J., Windeatt, T. (1996c). « Reliable surface reconstruction from multiple range images », *Fourth International European Conference on Computer Vision*, Cambridge, UK, Vol. 1, pp. 117-126.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W. (1992). « Surface reconstruction from unorganized data points », *Computer Graphics, Siggraph 92*, Chicago, USA, July, Vol. 26, pp. 71-78.

Lorensen, W. E., Cline, H. E. (1987). « Marching cubes: A high resolution 3D surface reconstruction algorithm », *Computer Graphics, Siggraph 87*, Anaheim, USA, July, Vol. 21, pp. 163-169.

Motavalli, S., Suharitdamrong, V., Alrashdan, A. (1998). « Design model generation for reverse engineering using multi-sensors », *Institute of Industrial Engineers (IIE) Transactions*, Vol. 30, pp. 357-366.

Nikolaidis, N., Pitas, I. (2001). « 3D image processing algorithms », *John Wiley & Sons*, New York, USA.

Pito, R. (1996). « Mesh integration based on co-measurements », *IEEE International Conference on Image Processing*, Lausanne, Switzerland, September, pp. 397-400.

Pressman, R.S. (1997). « Software engineering: A practitioner's approach », *McGraw-Hill Companies*, Columbus, USA.

Reed, M. K., Allen, P. K. (1997). « A robotic system for 3D model acquisition from multiple range images », *IEEE International Conference on Robotics and Automation*, Albuquerque, USA, April, pp. 2509-2514.

Ritter, G.X., Wilson, J.N. (1996). « Computer vision algorithms in image algebra », *CRC Press*, Boca Raton, USA.

Roth, G., Wibowo, E. (1995). « A fast algorithm for making mesh models from multi-view range data », *DND/CSA Robotics and Knowledge Based Systems Workshop*, St-Hubert, Canada, October, pp. 349-356.

Rutishauser, M., Stricker, M., Trobina, M. (1994). « Merging range images of arbitrary shaped objects », *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June, pp. 573-580.

Soucy, M., Laurendeau, D. (1995a). « A dynamic integration algorithm to model surfaces from multiple range views », *Machine Vision and Applications*, Vol. 8, No. 1, pp. 53-62.

Soucy, M., Laurendeau, D. (1995b). « A general approach to the integration of a set of range views », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, pp. 344-358.

Swokowski, E.W. (1991). « Analyse », *PWS-Kent Publishing Company*, Boston, USA.

Turk, G., Levoy, M. (1994). « Zippered polygon meshes from range images », *Computer Graphics, Siggraph 94*, Orlando, USA, July, Vol. 26, pp. 311-318.

Wheeler, M., Sato, Y., Ikeuchi, K. (1996). « Consensus surfaces for modelling 3D objects from multiple range images », *Tech. Rep., Carnigie Mellon Univ., School of Computer Science*, #CMU-CS-TR-96-168. <ftp://reports-archive.adm.cs.cmu.edu/1996>.

Wright, R.S., Sweet, M. (1996). « OpenGL superbible », *Waite Group Press*, Indianapolis, USA.