



HAL
open science

Réalisation d'un plug-in de manipulation d'images 3D au sein de l'environnement SolidWorks

Blandine Ganivet

► **To cite this version:**

Blandine Ganivet. Réalisation d'un plug-in de manipulation d'images 3D au sein de l'environnement SolidWorks. Sciences de l'ingénieur [physics]. 2014. dumas-01167499

HAL Id: dumas-01167499

<https://dumas.ccsd.cnrs.fr/dumas-01167499v1>

Submitted on 22 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS
ÉCOLE SUPÉRIEURE DES GÉOMÈTRES ET TOPOGRAPHES

MÉMOIRE

présenté en vue d'obtenir

le DIPLÔME D'INGÉNIEUR CNAM

Spécialité : Géomètre et Topographe

par

Blandine GANIVET

Réalisation d'un Plug-in de manipulation d'images 3D au sein de
l'environnement SolidWorks

Soutenu le 08 Juillet 2014

JURY

PRÉSIDENT : Monsieur Laurent MOREL

MEMBRES : Monsieur Lionel RAFFIN, maître de stage

Madame Elisabeth SIMONETTO, professeur référent

Monsieur Laurent POLIDORI

Monsieur Jérôme VERDUN

« L'art ne veut pas la représentation d'une chose belle mais la belle représentation d'une chose »
Emmanuel KANT

Avant-propos

Comme toute formation ingénieure, les trois années à l'ESGT aboutissent sur le Travail de Fin d'Études. Ce dernier permet de découvrir le type d'activité que peut rencontrer un jeune ingénieur lors de ses premiers pas dans le monde professionnel. De plus, il comprend un aspect recherche non négligeable qui représente un travail sur le long terme.

Ce travail a été élaboré dans le cabinet de géomètre-expert Geosat à CANEJAN en Gironde sur une période de cinq mois. Ce mémoire sera la retranscription de ces mois de recherches et d'analyse sur « La réalisation d'un plug-in de manipulation d'images 3D au sein de l'environnement SolidWorks » par le biais d'ortho-images créées par des logiciels de traitement de scans 3D.

Remerciements

Je tiens à remercier dans le cadre de mon TFE :

- Tout d'abord mon maître de stage, M. Lionel RAFFIN qui m'a fait confiance dans la réalisation de ce projet. Au fil de ces vingt semaines, M. RAFFIN a toujours répondu présent pour m'aider et me suivre dans l'évolution de mes recherches.
- Puis les deux autres géomètre-experts associés du cabinet, M. Cédrik FERRERO et M. Mathias SAURA pour m'avoir pris dans leur cabinet et pour leur attention tout au long de mon TFE.
- Pour son aide précieuse et sa disponibilité, M. Benjamin THOMAS.
- L'ensemble des employés du cabinet pour leur accueil chaleureux durant ces vingt semaines passées avec eux.
- Mon professeur référent Mme Élisabeth SIMONETTO pour son soutien, son aide et ses remarques constructives.

Dans le cadre de mes études, je souhaite remercier ma famille qui a toujours été là pour moi et tout particulièrement ma maman pour son soutien inéluctable durant ces années d'études

Table des matières

Introduction	Page 7
Chapitre I. Définition d'une image contenant l'information d'altitude	9
<u>1.1 Création d'une ortho-image à partir de nuage de points</u>	<u>9</u>
1.1.1 Définition d'une ortho-image	9
1.1.2 Procédure de création d'ortho-image à partir d'un nuage de points	9
1.1.2.1 <i>Le levé sur le terrain</i>	9
1.1.2.2 <i>Assemblage et nettoyage</i>	9
1.1.2.3 <i>Génération d'ortho-images</i>	9
1.1.2.4 <i>Le géoréférencement</i>	10
<u>1.2 Définition de la 2.5D</u>	<u>10</u>
1.2.1 Notion de dimension	10
1.2.2 La 2.5D	10
1.2.3 La différence entre une modélisation en 2.5D et en 3D	11
1.2.4 Pourquoi la 2.5D dans notre problématique	12
<u>1.3 Création d'une image solide</u>	<u>12</u>
1.3.1 Qu'est-ce qu'une image solide ?	12
1.3.1.1 <i>Structure d'une image solide</i>	13
1.3.1.2 <i>Gexcel et l'image solide</i>	13
1.3.2 Exemple d'utilisation avec le plug-in AutoCAD	15
1.3.2.1 <i>Création de l'image solide dans PointR3®</i>	15
1.3.2.2 <i>Utilisation du plug-in dans SolidWorks</i>	15
<u>1.4 Représentation de l'altitude dans l'image</u>	<u>16</u>
1.4.1 L'altitude dans une carte	16
1.4.2 Solution choisie pour répondre à la problématique	17
1.4.2.1 L'échelle	17
1.4.2.2 Le mode	18
Chapitre II. Processus d'importation des ortho-images	19
<u>2.1 Génération d'ortho-images à l'aide de différents logiciels</u>	<u>19</u>
2.1.1 PointCab	19
2.1.2 Scene FARO	19
2.1.3 Gexcel R3	19
<u>2.2 Présentation de l'interface du plug-in</u>	<u>20</u>
<u>2.3 Les éléments nécessaires pour l'importation</u>	<u>21</u>
2.3.1 Les formats d'image : JPEG et BITMAP	21
2.3.2 Application à la problématique	22
<u>2.4 Le processus de création</u>	<u>22</u>

2.5 Comment gérer les ortho-images dans le plug-in	24
2.5.1 Gérer les ortho-images	24
2.5.2 Sauvegarde des ortho-images	25
2.5.2.1 Qu'est-ce que le XML	25
2.5.2.2 Pourquoi utiliser le XML	25
2.5.2.3 Exemple de fichier XML	26
2.5.2.4 Application du XML au cas des ortho-images	27
2.6 Exemple d'utilisation	28
Chapitre III. Manipulation d'une image solide générée par R3	29
3.1 Récupération de l'information d'altitude dans l'image	29
3.1.1 Changement de système de coordonnées	29
3.1.1.1 La translation : le changement d'origine	29
3.1.1.2 Les effets de la rotation des axes	29
3.1.1.3 La combinaison de la translation et la rotation	30
3.1.1.4 Le changement de repère en 3D	30
3.1.2 Algorithme de Bresenham	31
3.1.2.1 Présentation	31
3.1.2.2 Fonctionnement de l'algorithme	32
3.1.3 Récupération de l'altitude	34
3.1.4 Le projeté orthogonal	34
3.1.4.1 Equation d'une droite	34
3.1.4.2 Projeté orthogonal	34
3.1.5 Retour dans le système de coordonnées général de SolidWorks	35
3.1.6 Dessin en 3D	35
3.1.6.1 L'ortho-image dans SolidWorks	35
3.1.6.2 Exemple d'utilisation	36
3.2 Création d'un MNT	38
3.2.1 Qu'est-ce qu'un MNT	38
3.2.2 De l'image solide à un MNT	39
3.2.3 La triangulation de Delaunay	40
3.2.3.1 Définition	40
3.2.3.2 Théorème de l'angle inscrit	41
3.2.3.3 Diagramme de Voronoï	41
Conclusion	43
Table des illustrations	45
Bibliographie et références web	47
Table des annexes	48

Introduction

Le 18 mars 2014, la ministre de l'Égalité des territoires et du Logement, Cécile Duflot annonce dans un entretien au Moniteur : « *Il faut déployer la maquette numérique(...) que nous allons rendre progressivement obligatoire dans les marchés publics en 2017(...) avec pour objectif d'avoir une vraie carte vitale du bâtiment* ». Cette maquette numérique est ce qu'on appelle le BIM (Building Information Modelling). Ce projet est développé avec l'objectif de contribuer à une réduction des énergies dans la construction et l'exploitation des complexes immobiliers. Le secteur du bâtiment présente un immense potentiel en ce qui concerne l'économie d'énergie, en effet il en est le plus gros consommateur avec plus de 45% des consommations énergétiques françaises. Afin de parvenir à réaliser ses économies la maquette numérique BIM sera un précieux atout.

Ainsi la modélisation 3D commence à prendre une part de plus en plus importante et à se développer dans de nombreux domaines. En effet, la réalisation de modèles en trois dimensions séduit une nouvelle clientèle. Elle est aujourd'hui présente dans des secteurs bien différents comme la construction, la topographie, l'industrie, l'archéologie, les jeux vidéos et bien d'autres.

La société Geosat, qui héberge ce travail de fin d'études, a vu le jour en janvier 2000 avec M.Cedrik FERRERO et elle est depuis quatorze années un cabinet de géomètre toujours à la pointe des dernières technologies afin de pouvoir répondre au plus grand nombre de demandes des clients. Aujourd'hui la société est dirigée par quatre associés (M. Cedrik FERRERO, M. Mathias SAURA, M. Lionel RAFFIN et M. Arnaud MATISSON) et possède deux bureaux en Gironde (Canéjan et Libourne) puis deux autres en Ile de France (Montrouge et Issy-les-Moulineaux).

A travers les différents cabinets les activités sont très variées : foncier, copropriété, topographie,... et développement d'un pôle 3D à Canéjan qui est le lieu d'accueil de ce travail de fin d'études. Afin de persévérer dans cette perspective avant-gardiste, la société Geosat possède quatre scanners laser FARO ce qui leur permet de pouvoir répondre à des demandes bien spécifiques.

En 2013, la société décroche le marché de modélisation 3D des bâtiments d'Airbus à Nantes qui ont une surface totale d'environ 40 000m². L'une des demandes de la société aéronautique est d'avoir cette représentation au format SolidWorks. En effet, SolidWorks est un logiciel plus connu dans l'industrie et la mécanique que dans le monde de la topographie (Geosat étant à notre connaissance le seul cabinet de géomètre en France à avoir une licence pour ce logiciel). Pour pallier à cette demande, les personnes en charge du dossier ont dû développer tout un processus de traitement des données bien spécifique. Afin d'importer les données 3D dans le logiciel demandé, il était nécessaire de passer préalablement par deux autres logiciels pour les traiter. Ce processus étant long, il est facile d'imaginer que le temps de traitement augmente rapidement avec un bâtiment de ces dimensions.

C'est dans ce contexte que le projet d'un développement en interne voit le jour pour améliorer ce processus et pouvoir traiter directement les données dans le logiciel SolidWorks à l'aide d'un plug-in (module d'extension d'un logiciel). Pour cela un travail de fin d'études est mis en place :

« Réalisation d'un plug-in de manipulation d'image 3D au sein de l'environnement SolidWorks »

Deux objectifs se dégagent donc pour ce projet, le second étant dépendant de la réussite du premier. Tout d'abord l'insertion d'images dans le logiciel SolidWorks. Après le traitement des scans, des ortho-images sont créées. Il est nécessaire de pouvoir les intégrer directement dans le logiciel sans passer par d'autres car cela augmente les manipulations et donc la possibilité de commettre des erreurs. L'enjeu principal étant de parvenir à une automatisation du processus d'insertion. Cela entraînera un gain de temps de traitement des dossiers.

Puis dans un second temps, une fois l'image insérée dans le logiciel, il serait souhaitable de pouvoir automatiser aussi le dessin en 3D à l'aide d'une image qui contiendrait l'information d'altitude. Il n'est pas envisagé de créer un nouveau format d'image mais bien de s'appuyer sur ce que nous fournissent les logiciels de traitements de scans existants. Pour cela la société italienne Gexcel ayant déjà développé un plug-in pour AutoCAD sera notre partenaire tout au long de cette étude.

Reprenant ces deux enjeux, ce mémoire décrira dans un premier temps comment sera définie l'image contenant l'information d'altitude, qu'elles sont ses propriétés, comment elle est créée ainsi que la représentation de l'altitude dans l'image. Puis une seconde partie sera consacrée au processus d'insertion d'images dans l'environnement SolidWorks grâce au plug-in et comment elles sont gérées. Un exemple de réalisation de modélisation 3D permettra de mieux se rendre compte de l'utilisation. Enfin, la dernière partie exposera les différentes fonctions d'automatisation de dessin 3D que propose le plug-in pour aboutir à un volume ou à la création d'un MNT.

Chapitre I Définition d'une image contenant l'information d'altitude

1.1. Création d'une ortho-image à partir de nuages de points

1.1.1 Définition d'une ortho-image

Qu'est-ce qu'une ortho-image et comment est-elle formée ? Voici la définition donnée dans le Bulletin d'Information de l'IGN n°74 (2003/03) par Didier Boldo chercheur à l'IGN :

"Une ortho-image est une image, aérienne ou satellitaire, superposable à une carte. Chaque pixel est ainsi localisé par des coordonnées géographiques. Une ortho-image est donc une image déformée de telle manière que le centre de prise de vue soit à la verticale de chaque point : cette image est en perspective axonométrique verticale. Dans les ortho-images classiques les défauts corrigés sont :

- la distorsion de l'objectif,
- la non verticalité de l'axe de prise de vue,
- la rotondité de la terre,
- le relief du sol naturel.

Les « vraies ortho-images » sont des ortho-images où tous les défauts, y compris les dévers, ont été corrigés. Tous les objets, bâtiments compris, sont donc à leur position cartographique. Pour calculer ces vraies ortho-images, il est nécessaire de disposer de modèles de la surface topographique, appelés Modèles Numériques de Surface (MNS)."

1.1.2 Procédure de création d'ortho-images à partir d'un nuage de points

Les différentes étapes nécessaires pour générer une ortho-image à partir d'un nuage de points sont :

1) **Le levé scan sur le terrain** : cette première phase représente l'acquisition des mesures sur le terrain par technique de lasergrammétrie (voir figure 1). Plusieurs scans peuvent être nécessaires pour atteindre tous les points à mesurer. C'est pourquoi suit l'étape suivante qui est l'assemblage et le nettoyage.

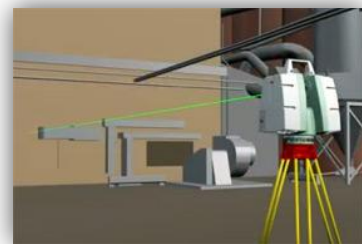


Figure 1 : Levé scan (leica-geosystems.fr)

2) **Assemblage et nettoyage** : passage essentiel pour avoir la précision finale du document. Le but étant d'effectuer un traitement des fichiers de mesures provenant des levés de terrain pour générer un nuage de points global. Afin de pouvoir assembler au mieux les différents scans, il est préférable d'utiliser des cibles spécifiques qui seront reconnues par le logiciel de traitement 3D.

3) **Génération d'ortho-images** : après l'unification des scans, il est possible de projeter des points colorisés que l'on définit dans une certaine épaisseur sur un plan 2D, le résultat est une ortho-image.

4) **Le géoréférencement** : chaque scan est géolocalisé grâce à des relevés de géomètre afin d'obtenir des fichiers de mesures connues en coordonnées. Cela permet aussi d'avoir les informations de géoréférencement des ortho-images créées précédemment et de pouvoir les assembler.

1.2 Définition de la 2.5D

1.2.1 Notion de dimension

Avant tout, il est nécessaire de définir qu'est-ce qu'une dimension. C'est une composante qui permet de décrire le monde réel.

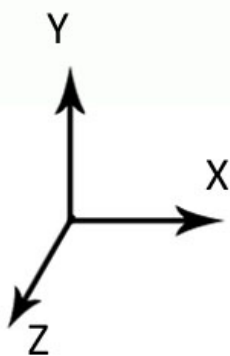


Figure 2 : trièdre direct

Notre monde est composé de 4 dimensions (que nous abrégons en 4D) malgré que plus communément nous pensons qu'il en est constitué que de trois. Grâce à notre acuité visuelle nous sommes capables de discerner trois facteurs : le premier horizontal, nommé habituellement X. Une seule dimension permet de définir seulement un axe et il est donc impossible de représenter une quelconque forme. D'où la nécessité d'une deuxième dimension que l'on appelle Y. Il devient donc possible de se déplacer horizontalement et verticalement et d'afficher des formes planes, comme des cercles, des rectangles ou plus concrètement des dessins, des photos,... La troisième dimension (Z) régit la dernière possibilité de déplacement : la profondeur. Avec cette ultime dimension, les déplacements dans toutes les directions deviennent donc possible, comme dans le monde réel.

Les trois axes ensemble forment un trièdre direct comme représenté figure 2.

Mais qu'elle est-elle donc cette quatrième dimension dont nous avons parlé précédemment ?

La notion de quatrième dimension émerge dans l'Encyclopédie de d'Alembert : *"Un homme d'esprit de ma connaissance croit qu'on pourrait cependant regarder la durée comme une quatrième dimension"*. Cette idée se retrouve aussi en physique avec la relativité d'Einstein où l'espace-temps représente un seul et même référentiel.

Nous retrouvons ce concept de quatrième dimension dans les logiciels de modélisation. En effet, il est possible d'animer les objets représentés en 3D en fonction du temps.

1.2.2 La 2.5D

Nous avons défini ce qu'est une dimension, mais qu'est-ce que cette demie dimension et à quoi correspond-elle ?

La 2.5D voit le jour au cœur des années 1970 dans le milieu des jeux vidéo. Les premiers à l'utiliser sont les jeux d'arcades dont par exemple les jeux de simulation de combat aérien.



Figure 3 : Super Mario Bros (giantbomb.com)

La notion de profondeur apparaissait grâce à l'augmentation ou la diminution de la taille de l'avion ennemi suivant si on se rapproche ou on s'éloigne. Aujourd'hui de nombreux jeux vidéo sont encore en 2.5D, exemple ci-dessus de Super Mario Bros (figure 3).

La 2.5D est donc une "pseudo 3D" qui désigne les technologies d'imagerie qui se trouvent à mi-chemin entre la 2D et la 3D. La raison principale de l'utilisation de cette technique plutôt que de la "vraie 3D" est que ça demande beaucoup moins de traitement par un système informatique pour l'affichage des graphiques.

La méthode utilisée pour créer l'illusion d'être dans un environnement en trois dimensions est d'étirer une image pour qu'elle devienne plus grande afin de donner l'effet que l'objet représenté arrive et se rapproche de l'utilisateur. Cette technique est réalisable seulement si l'image est vue depuis un point fixe. Ainsi il n'est pas possible de parler de 2.5D si on souhaite faire une rotation de notre objet.

1.2.3 La différence entre une modélisation en 2.5D et en 3D

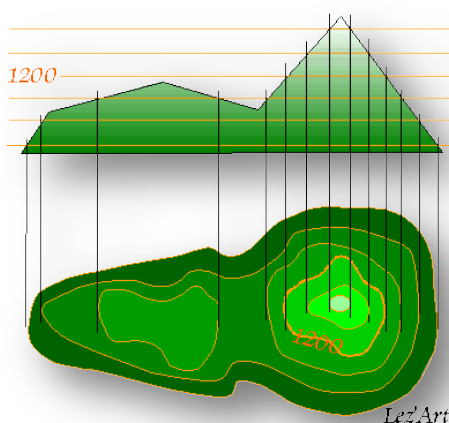


Figure 4 : Courbes de niveau (lezart.free.fr)

Une modélisation en 2.5D implique qu'à chaque point du sol défini par ces coordonnées planaires (x,y) correspond une seule valeur de z. Nous pouvons reformuler cela mathématiquement avec l'équation suivante : $z=f(x,y)$. Un exemple pour illustrer l'utilisation de la 2.5D est les courbes de niveau (figure 4).

Les courbes de niveau sont des lignes imaginaires qui joignent tous les points d'un terrain se trouvant à la même altitude.

Chaque point de la courbe ne peut avoir qu'une seule valeur d'altitude. Contrairement à la 3D vraie qui permet d'avoir des altitudes différentes pour une même paire de coordonnées (x,y). Prenons l'exemple d'un pont, il ne sera pas possible en 2.5D d'affecter les valeurs du dessus et du dessous du pont. Plus généralement, la 3D permet la modélisation intérieure (de bâtiments, de sous-sols, de cavités) que la 2.5D ne permet pas.

Pour compléter l'analyse il est aussi possible de faire une différence entre la 2.5D et la 2D+H. Comme expliqué précédemment en 2.5D chaque point a une valeur unique de z mais chaque point peut avoir une valeur de z différente donc le relief est défini avec la précision qu'il y a entre deux points. La 2D+H est utilisée lorsqu'on extrude un polygone avec une valeur unique de hauteur. Voir figure 5 : BD TOPO® - représentation 3D du Havre.

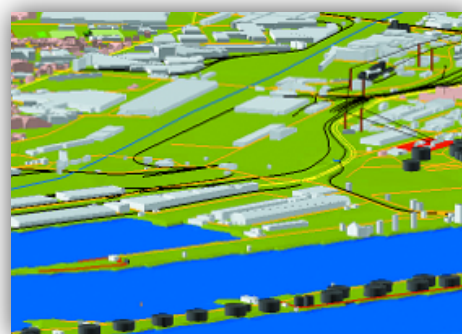


Figure 5 : BD TOPO 3D du Havre (ign.fr)

1.2.4 Pourquoi la 2.5D dans notre problématique?

Rappelons que la problématique est de dessiner dans le logiciel SolidWorks en 3D à l'aide d'ortho-image. Une image est une matrice composée d'un nombre fini de pavés que l'on appelle des pixels (figure 6). Chaque pixel représente un élément de la matrice, il a un identifiant unique qui est l'indice de la ligne et de la colonne (figure 7).

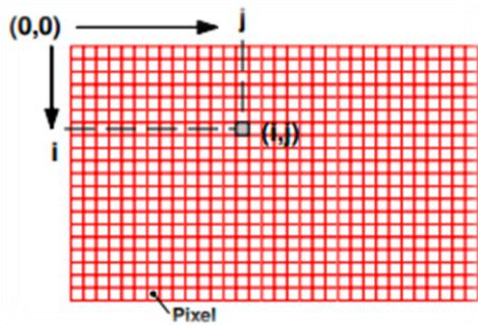


Figure 6 : Représentation matricielle d'une image



Figure 7 : Image matricielle (wikipédia.fr)

Le principe de la solution envisagée est d'utiliser une ortho-image comme base qui permet de connaître la position d'un pixel en x, y puis d'associer une valeur d'altitude (z) à chacun de ces pixels. Les coordonnées d'un pixel dans l'image ont pour origine le coin en haut à gauche de l'image.

Ainsi il n'est donc pas possible de pouvoir dessiner en "vraie" 3D car chaque pixel ne peut avoir qu'une seule information d'altitude. La fonction de dessin développée permettra d'avoir un modèle en 2.5D et ne pourra pas gérer les z multiples qu'on peut rencontrer dans le monde réel.

1.3Création d'une image solide

1.3.1 Qu'est-ce qu'une image solide

Le concept d'image solide a été présenté en septembre 2002 en Grèce lors d'un meeting de l'IPRS (International Society for Photogrammetry and Remote Sensing). D'après l'étude menée par Leandro Borna et Sergio Dequal (voir annexe 1), en connaissant l'orientation interne et externe d'une caméra et en considérant une image comme étant la projection d'un objet photographié, il est possible, pour chaque pixel de l'image de déterminer son axe de projection dans l'espace.

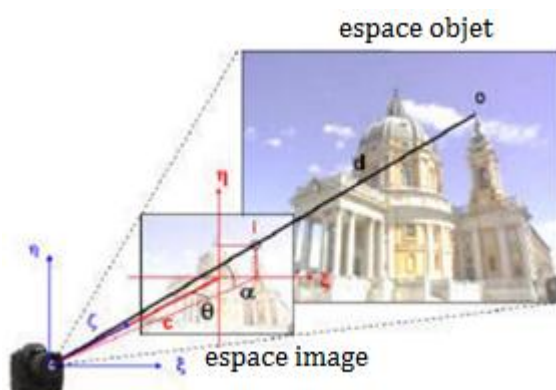


Figure 8 : Relation entre l'espace image et objet (voir annexe 1)

Mais une image n'est pas suffisante pour déterminer les coordonnées 3D d'un point. En effet, la direction doit être associée à une distance sinon tous les points positionnés sur cette direction appartiennent au même pixel de l'image (figure 8).

L'intégration des données de levés scanner laser dans les données de l'image offre la possibilité d'associer une distance pour toutes les directions définies par chaque pixel. Cette distance est celle qui relie le centre de prise de vue de l'appareil et l'objet à modéliser. Cette distance avec la direction dans l'espace donnée par deux angles nous ramène à un système polaire 3D. Une image avec le système de radiométrie RVB associée avec les données d'un scanner laser forment ce nouveau concept : une image solide.

1.3.1.1 Structure d'une image solide

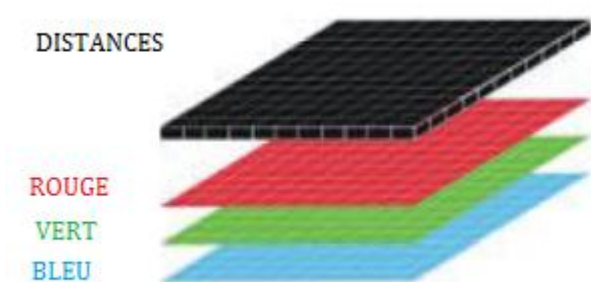


Figure 9 : Structure d'une image solide (voir annexe 1)

Les distances qui proviennent des données scanner sont stockées dans une matrice qui a les mêmes dimensions que les trois matrices R, V et B. Ainsi une image solide est composée de quatre niveaux de matrices : RVB et la distance (figure 9).

Pour les matrices RVB, chaque pixel est codé sur 1 octet alors que les pixels de la matrice des distances sont codés sur 2 ou 4 octets.

1.3.1.2 Gexcel et l'image solide



Gexcel a vu le jour en 2007 et se donne pour mission de transformer les dernières avancées scientifiques dans le domaine de la géomatique en produits et services de haute technologie dans des domaines divers. Le but étant de fournir aux entreprises clientes des améliorations significatives de leur efficacité.

Reprenant le concept de l'image solide, la société italienne a développé un logiciel qui permet de créer des images solides. "Gexcel Solid Image" est le résultat d'un outil spécialisé appelé "Virtual Scan" qui est inclus dans le logiciel développé PointR³®. Il crée une vue en perspective ou orthographique du nuage de point 3D obtenu dans PointR³®. La vue en perspective ou orthographique est définie par une caméra et un angle de vue que l'on appelle tronc (frustrum).

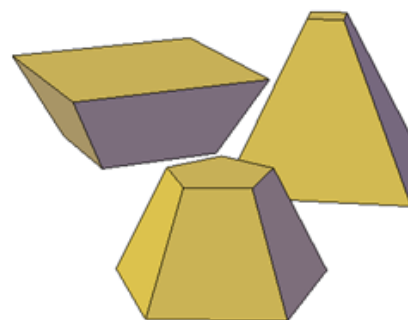


Figure 10 : Exemple de tronc (wikipedia.fr)

Le tronc est une portion d'un solide qui lie et coupe deux plans parallèles (figure 10).

Lorsqu'on a une vue en perspective, le tronc sera une pyramide (figure 12) alors qu'avec une vue orthographique ce sera un parallélépipède (figure 11).

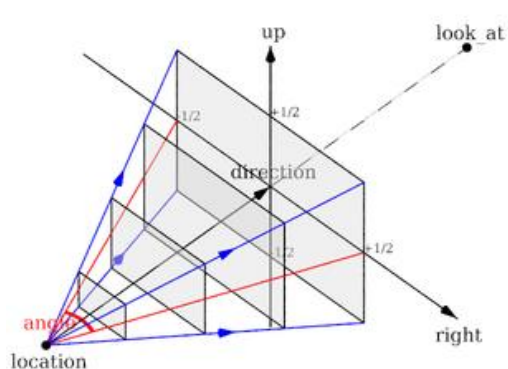


Figure 12 : Vue en perspective (wiki.povray.org)

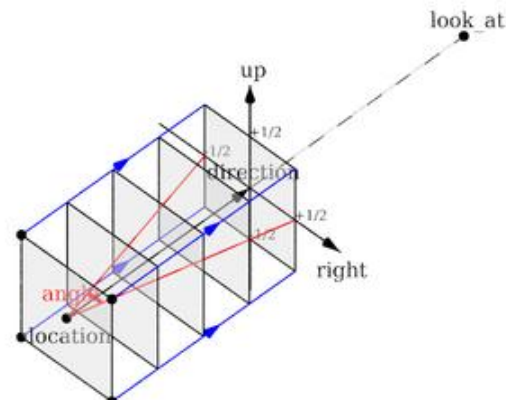


Figure 11 : Vue orthographique (wiki.povray.org)

L'outil "Virtual Scan" permet de définir ce que nous appellerons ici la résolution (la taille du pixel en mètre) que nous souhaitons obtenir en fonction de la taille de la caméra (largeur et hauteur) et de la profondeur d'acquisition (dépendante de la distance entre les deux plans parallèles du tronc). Grâce à cet outil il est possible d'obtenir plusieurs résultats :

- une ortho-image (uniquement dans le cas d'une vue orthographique),
- un nouveau nuage de points,
- une image solide,

Cette image solide est formée dans un fichier spécifique défini par Gexcel. Pendant le scan virtuel, la scène 3D que nous avons choisie est capturée dans un lot d'images : une pyramide d'images (figure 13) qui ont des résolutions différentes. Généralement la pyramide a trois niveaux.

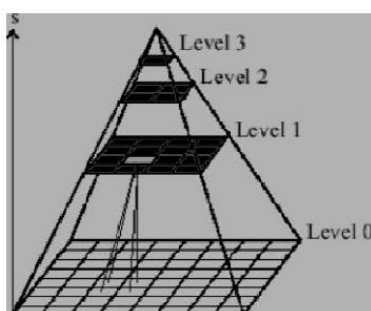


Figure 14 : Pyramide d'image (Gexcel)



Figure 13 : Images de la pyramide (Gexcel)

Au premier niveau, l'image a la plus grande résolution et les suivantes sont définies avec 1/2 et 1/4 de cette résolution (figure 14).

Chaque image est ensuite divisée suivant un quadrillage pour former un lot de sous-image. Ceci permettant de préserver pour chaque portion la résolution désirée. Cette subdivision est utile à certains logiciels qui ne peuvent gérer et afficher que des images avec une certaine taille.

Pendant l'enregistrement de l'image solide chaque portion est stockée en .png et un nom lui est affecté automatiquement en fonction du niveau de la pyramide auquel elle appartient. Exemple pour le niveau 0 : image_0_0_0, niveau 1 : image_1_0_0 puis image_1_1_0 etc. Pour chacune de ces portions les informations relatives à la caméra utilisée et la distance entre la caméra et la scène 3D sont stockées. Toutes ces données sont sauvegardées dans un fichier compressé. Ce fichier peut être lu directement dans une visionneuse spécialisée contenue dans PointR³® qui permet de zoomer et dézoomer dans la pyramide d'image, de lire les coordonnées d'un point et mesurer la distance entre deux points spécifiés.

Un plug-in pour AutoCAD a été développé par Gexcel pour pouvoir afficher les portions d'image avec la plus haute précision et permettre la manipulation dans le logiciel de DAO.

1.3.2 Exemple d'utilisation avec le plug-in AutoCAD

1.3.2.1 Création de l'image solide dans PointR3®

Après avoir importé un nuage de points dans PointR3®, le logiciel nous offre la possibilité de créer une ortho-image ou une "image solide" grâce à la définition d'un plan vertical ou horizontal. Il est aussi possible de les créer à partir de la vue courante.

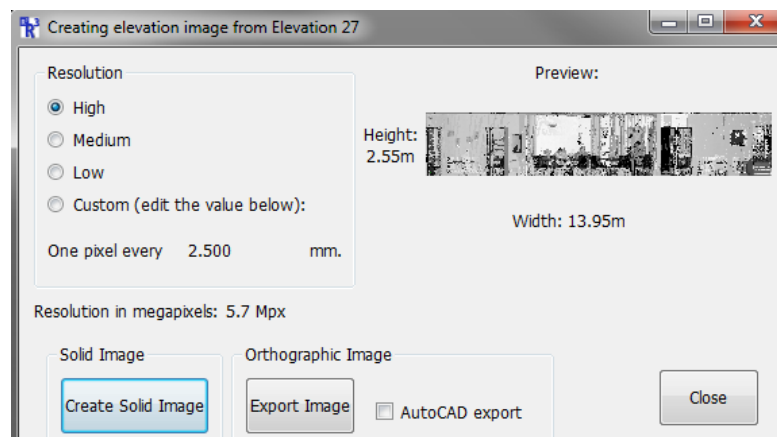


Figure 15 : Paramétrage d'export (Gexcel)

Il est possible de choisir la résolution que l'on souhaite (figure 15) :

- haute, un pixel mesure 2,5mm.
- moyenne, un pixel mesure 5mm.
- basse, un pixel mesure 10mm.
- enfin la possibilité d'indiquer soi-même la valeur que l'on souhaite pour la taille d'un pixel.

1.3.2.2 Utilisation du plug-in dans SolidWorks

Une fois l'image solide créée, nous obtenons un fichier .r3s un format de fichier créé pour les "solides images". Après l'installation du plug-in dans AutoCAD, ces fichiers sont compatibles et peuvent être utilisés. Une fonction pour dessiner en 3D doit être validée et voici ce qu'on peut obtenir : (figure 16)

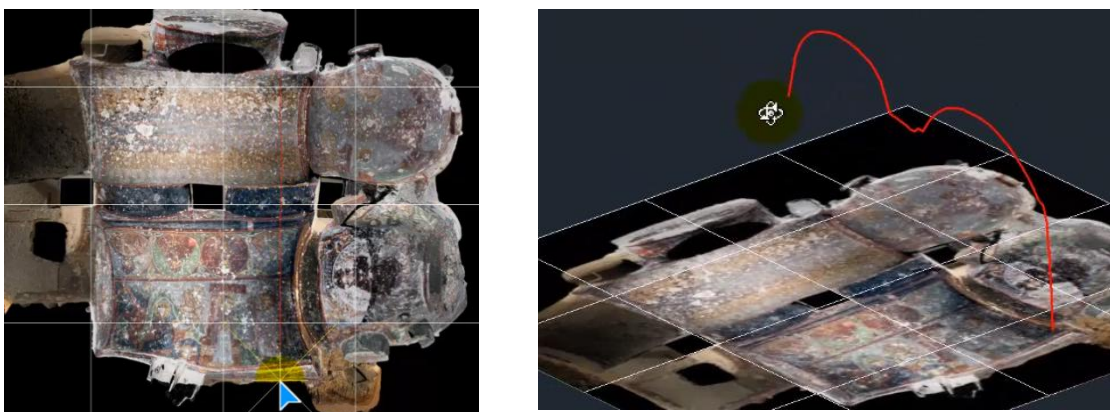


Figure 16 : Exemple utilisation plug-in AutoCAD (Gexcel)

1.4 Représentation de l'altitude dans l'image

1.4.1 L'altitude dans une carte

Le format d'échange .r3s étant bien spécifique à l'utilisation du plug-in d'AutoCAD, il a fallu réfléchir à un autre moyen de stocker l'information d'altitude dans l'image. Il existe déjà plusieurs formes de représentation de l'altitude dans une carte.

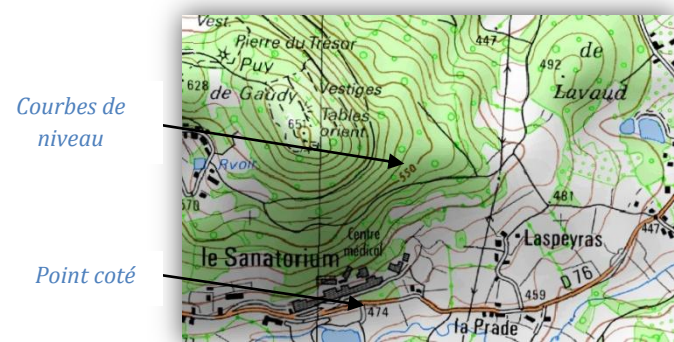


Figure 17 : Exemple carte (loisir.ign.fr)

Reprenons l'exemple des courbes de niveau cité précédemment. L'altitude des courbes maitresses est indiquée, puis elles sont équidistantes de 10m (figure 17). A l'aide des courbes de niveaux il est possible de dessiner des profils en long entre deux points déterminés sur la carte.

Il est quand même bon de notifier que cette représentation s'adresse à un public initié à la lecture d'une carte.

En complément des courbes de niveau, sur certaines cartes il est fréquent de rencontrer une autre représentation de l'altitude. Celle-ci ne nous indique pas une valeur précise d'altitude mais permet une meilleure compréhension de la carte, il s'agit de l'estompage.

L'estompage (figure 18) est la partie plus sombre qu'on aperçoit sur les cartes comme ci-contre. Elle symbolise l'ombre du relief éclairé par un soleil fictif se trouvant à 40° au-dessus de l'horizon côté Nord-Est.

Afin de faciliter la lecture des courbes de niveau, les lieux remarquables tels que les sommets, les cuvettes ou les changements de pente sont marqués par un point coté.

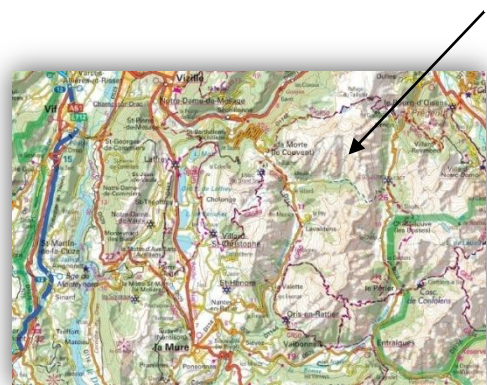


Figure 18 : Exemple carte (loisir.ign.fr)

Un autre exemple de représentation altimétrique qui peut être en complément des deux premiers et l'utilisation de la couleur. La variation d'altitude peut être représentée en jouant sur différents facteurs : la teinte, la luminosité et la saturation. (figure 19)

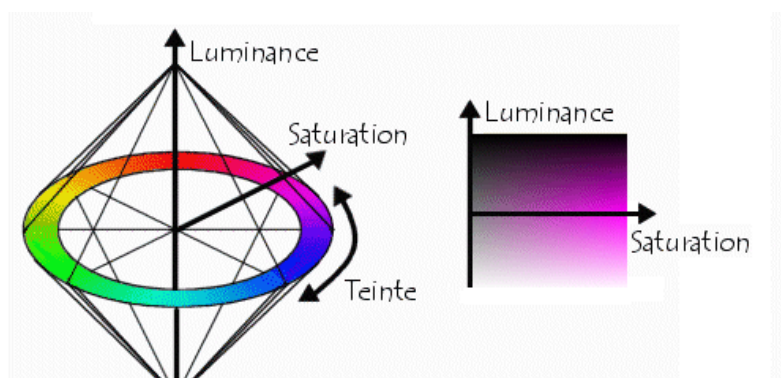


Figure 19 : Système TLS (commentcamarche.net)

TLS (Teinte, Luminosité, Saturation) est un système colorimétrique. Il sert à décrire toutes les couleurs que l'homme peut distinguer (qui ont une longueur d'onde comprise entre 400 et 800 nm). La teinte est définie par un angle compris entre 0 et 360° alors que la luminosité et la saturation se définissent en pourcentage.

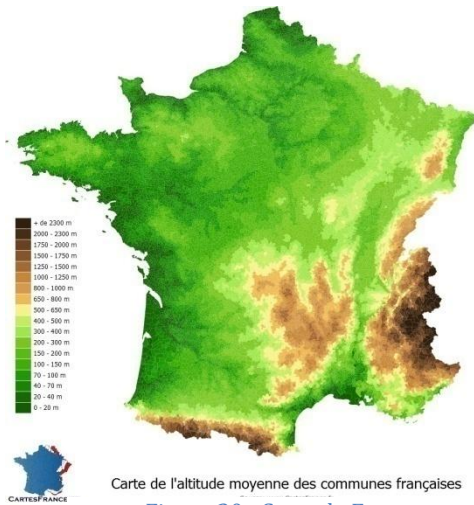


Figure 20 : Carte de France
(cartefrance.fr)

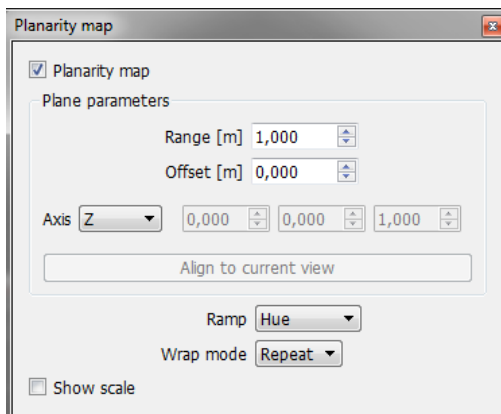
Voici un exemple de représentation altimétrique des communes françaises grâce à une variation locale de teinte allant du vert foncé (0-20 m) jusqu'au marron (+2300 m) en passant par un dégradé de couleurs (voir figure 20). De même, cette variation peut être faite avec la luminosité ou la saturation.

Cette représentation est plus explicite que les courbes de niveau. Il est quand même nécessaire d'avoir l'échelle correspondante aux différentes teintes pour pouvoir donner une valeur d'altitude.

1.4.2 Solution choisie pour répondre à la problématique

Ne pouvant pas employer le format généré lors de la création d'une image solide, il est donc nécessaire d'utiliser les différentes options d'export d'ortho-image pour concevoir notre propre image solide.

Le logiciel PointR3® offre la possibilité de créer une carte planimétrique. C'est une ortho-image dont la variation d'altitude est représentée grâce à la variation de couleurs.



Les paramètres sont : (figure 21)

- les valeurs maximales et minimales qu'on souhaite représenter.
- l'axe suivant lequel est fait l'étalement c'est-à-dire suivant quel axe va être définie la variation de couleur.
- l'échelle (voir 1.4.2.1) peut être matérialisée suivant la teinte ou la variation de niveau de gris.
- un mode étendu, répétition ou miroir (voir 1.4.2.2).

Figure 21 : logiciel PointR3, planarity map

1.4.2.1 L'échelle

Sachant que nous cherchons à obtenir une image qui contient les valeurs d'altitude de chaque pixel, l'étalonnage se fera suivant l'axe Z. Puis se pose la question de la matérialisation de l'échelle et le mode. En ce qui concerne la représentation de l'échelle que ce soit la teinte ou la variation de niveau de gris ne présente pas une différence majeure à part la visualisation bien évidemment mais aussi l'étalement des valeurs. En effet lorsqu'on travaille avec la teinte l'échelle est la suivante :




0° équivaut donc à notre valeur minimale et 360° à la valeur maximale.

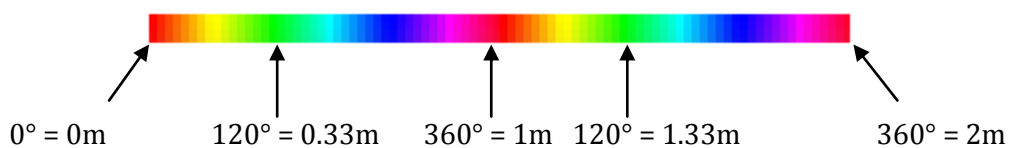
Avec la variation de niveau de gris, il est possible d'avoir une échelle de valeurs entières allant de 0 pour l'altitude minimale et 255 pour la maximale. Ces valeurs proviennent du fait que la composante est codée sur 8 bits.

Le bit étant l'unité de codage informatique. Elle ne peut prendre que deux valeurs : vrai ou faux qui équivaut à 1 ou 0 en informatique. Il existe donc deux possibilités de résultats et pour savoir le nombre de possibilité pour 8 bits le calcul est le suivant : $2^8 = 256$ niveaux de gris.

L'avantage d'utiliser la teinte est d'avoir un nombre de possibilités de valeurs d'altitude supérieur à celle avec la variation de niveau de gris, ce qui entraîne une plus grande précision. Avec la variation de teinte il est possible d'avoir 360 valeurs donc 1° représente une variation de 8mm alors qu'entre chaque niveau de gris il y aura 12mm.

1.4.2.2 Le mode :

Les deux modes présentent quant à eux une grande différence. En voici une démonstration : admettons que nous avons choisi le travailler avec la teinte en mode répétition, la valeur minimale sélectionnée étant 0m et la maximale 1m. Imaginons que nous cherchons à connaître l'altitude d'un pixel dont la teinte est :  → 120°.



En mode répétition, l'échelle boucle sur elle-même. Donc si nous définissons que l'ortho-image prend en compte tous les points du nuage de points se trouvant dans une épaisseur de 3m, il sera impossible de savoir si un pixel ayant pour teinte 120° a une altitude de 0.33m, 1.33m ou 2.33m.

Le mode miroir se présente comme ceci : 

Pour les mêmes raisons que le mode répétition, le miroir ne répond pas aux attentes de notre problème.

Le mode étendu permet comme son nom le sous-entend d'étendre l'échelle aux valeurs maximale et minimale choisies. Afin d'avoir une unique valeur pour chaque altitude l'écart entre les deux valeurs maxima doit être égal à l'épaisseur du nuage de points qu'on souhaite décrire dans l'ortho-image.

Exemple avec une épaisseur de 3m où on souhaite modéliser le second étage d'un bâtiment :



Cette solution est donc celle choisie pour construire notre ortho-image contenant l'information d'altimétrie.

Chapitre II Processus d'importation des ortho-images

2.1 Génération d'ortho-images à l'aide de différents logiciels

2.1.1 PointCab



PointCab est une société allemande qui développe des logiciels de traitement de scans de levé 3D. L'avantage de ce logiciel est qu'il offre la possibilité d'être utilisé avec des scanners laser de différentes marques. En effet il est compatible avec FARO, Leica, Trimble et bien d'autres. Ce qui permet par la suite d'avoir un grand nombre de formats d'export pour s'adapter à un maximum de logiciel de CAO (Conception assistée par ordinateur).

2.1.2 Scene Faro

Faro est une société qui a été fondée en 1981 à Montréal au Canada. Elle va rapidement trouver une place majeure dans le monde de la mesure 3D. Son siège est situé en Floride à Lake Mary mais elle possède plusieurs filiales à travers le monde (Europe, Amérique du Sud et Asie). La société développe et commercialise dans logiciels et appareils de mesure et de numérisation assistés par ordinateur. Elle équipe de ces différents produits un grand nombre de petites, moyennes et grandes entreprises telles que : AIRBUS, AUDI, BOSCH,...

SCENE

2.1.3 Gexcel R3

GEXCEL Software Solutions

Présentée précédemment, la société italienne a été en partenariat avec Geosat tout au long de ce travail de fin d'études pour répondre au plus près de la problématique. En effet grâce à l'élaboration de leur image solide dans le logiciel PointR³®, il sera possible de dessiner en 3D directement à partir de l'image.

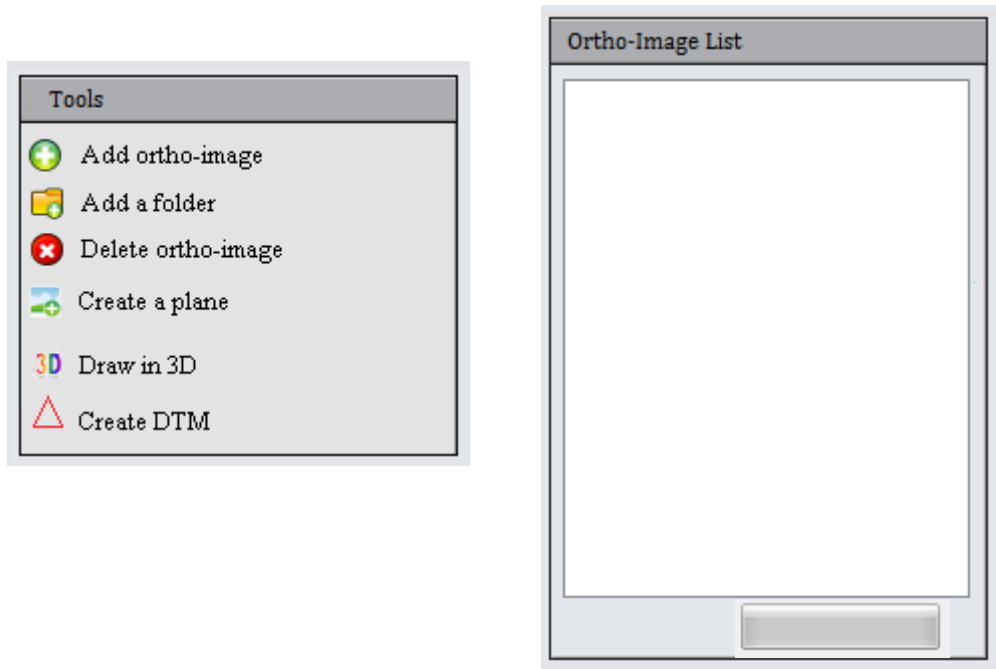



Le logiciel pour lequel cette société est particulièrement connue est JDC 3D Reconstructor (figure 22). En effet il permet de gérer des nuages de points et des images provenant de différentes plateformes d'imagerie et de scanners. Il est capable de combiner des données provenant de capteurs lidar terrestres, mobiles et embarqués comme des drones avec des données d'image 3D. (Voir Annexe 1)


Figure 22 : JRC 3D Reconstructot (gexcel.it)


2.2 Présentation de l'interface du plug-in


Le plug-in apparaît comme volet de tâche dans la fenêtre SolidWorks. Il présente plusieurs fonctionnalités présentées ci-dessous :




 Ajouter une ortho-image, peu importe qu'elle provienne de l'un des trois logiciels présentés ci-dessus. Il y a la possibilité d'ajouter une ou plusieurs ortho-images.


 Supprimer l'ortho-image sélectionnée dans la liste ci-contre.

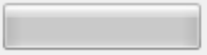
 Ajouter un dossier d'ortho-image. Elles seront toutes ajoutées au fur et à mesure dans le projet, dans la liste et dans l'arbre de création de SolidWorks.

 Création d'un plan. Lors de l'import des ortho-images afin de pouvoir dessiner il est nécessaire de créer un plan d'esquisse.

Les deux fonctions suivantes sont spécifiques aux ortho-images provenant de Gexcel car elles utilisent l'altitude de ces pseudos images solides.

 Convertir une ligne dessinée en 2D sur l'ortho-image en 3D. Elle offre aussi la possibilité d'extruder cette ligne pour obtenir un volume.

 Créer un MNT (Modèle Numérique de Terrain) à partir de points dessinés sur l'ortho-image.

 Une barre de chargement qui indique l'avancement de l'ajout des ortho-images dans SolidWorks.

Enfin, les ortho-images sont ajoutées dans une liste qui permet de les sélectionner pour pouvoir leur affecter les différentes fonctionnalités décrites ci-dessus.

2.3 Les éléments nécessaires pour l'importation

Afin de pouvoir importer une ortho-images dans SolidWorks le fichier image seul ne suffit pas. En effet, pour qu'elle soit intégrée dans le logiciel et géoréférencée il est nécessaire d'avoir un fichier qui contient les coordonnées de l'image. Cela permettra lors de l'importation de plusieurs ortho-images qu'elles soient automatiquement placées dans l'espace en 3D .

Ainsi il est indispensable d'avoir un fichier image (au format .jpg), un fichier texte contenant les coordonnées de l'image et une seconde image sera nécessaire pour récupérer l'information d'altitude. Ces deux fichiers contiennent la même image mais ne seront pas au même format. La première au format JPEG sera celle qui apparaîtra dans SolidWorks alors que la seconde sera en BITMAP et ne sera jamais affichée.

2.3.1 Les formats d'images : JPEG et BITMAP

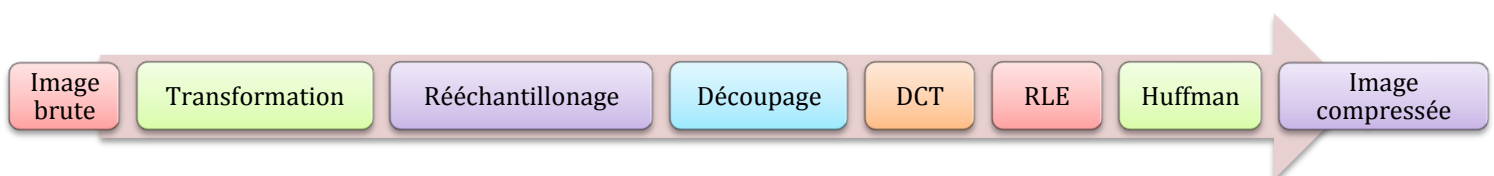
Bitmap est le format d'image le « plus brut ». Il est aussi appelé DIB (Device Independent Bitmap). Un fichier .bmp contient les données suivantes :

- en-tête 1 : les informations sur le fichier (l'identificateur de format, la dimension du fichier)
- en-tête 2 : les informations sur les données (largeur et hauteur en pixels, bit/pixel,...)
- la palette de couleurs utilisée
- les données

L'avantage d'une image au format .bmp est qu'il ne diminue pas la qualité de l'image car il peut en principe être compressé mais ce n'est généralement pas utilisé.

Le format JPEG (Joint Photographic Experts Group, le groupe qui l'a inventé) quant à lui provient d'une technique de compression. C'est le format le plus approprié pour compresser et envoyer des images (par e-mail par exemple). Il est bon de prendre en compte que généralement chaque fois qu'un fichier est enregistré au format JPEG, une nouvelle compression se passe avec une perte d'information.

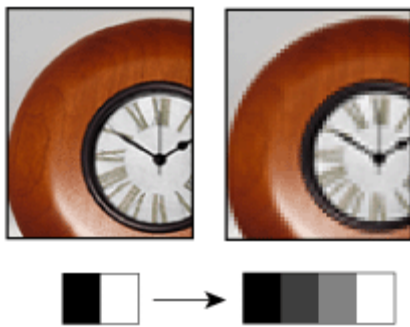
Les différentes étapes de la compression JPEG sont :



La transformation des couleurs : passage du système colorimétrique RGB (trois canaux Rouge, Vert, Bleu avec des valeurs de 0 à 255) à YCbCr. Y étant l'information sur la luminance et Cb et Cr les informations de chrominance. La conversion entre les deux systèmes colorimétriques est (classiquement) :

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

$$Cb = - 0.1687 * R - 0.3313 * G + 0.5 * B + 128 \quad Cr = 0.5 * R - 0.4187 * G - 0.0813 * B + 128$$



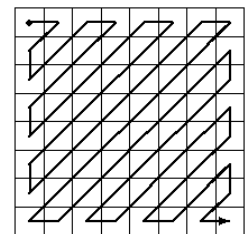
Le rééchantillonnage de la chrominance (figure 23): l'œil humain étant moins sensible à la chrominance qu'à la luminance, le principe est de sous-échantillonner ces canaux. L'inconvénient est que cela réduit la qualité de l'image et des artefacts apparaissent.

Figure 23 : Rééchantillonnage (zabaque.upac.ca/infographie)

Le découpage consiste à former des blocs de 8*8 pixels. Puis on leur applique la DCT (Discrete Cosinus Transform). Cette transformation mathématique permet de décomposer l'image en somme de fonctions cosinus. Elle se traduit mathématiquement :

$$DCT(i, j) = \frac{2}{N} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{pixel}(x, y) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2y+1)j\pi}{2N} \right]$$

La quantification est l'étape du procédé durant laquelle s'effectue la perte d'information. Pour chaque bloc de l'étape précédente, la quantification consiste à appliquer un coefficient de perte pour « supprimer » ou diminuer les mesures de hautes fréquences. Ce processus permettra de réduire les détails. Puis il examine le bloc avec un codage RLE (parcour en zigzag ayant pour effet de regrouper le plus de valeurs nulles possible)(figure244).



La dernière étape est l'encodage de l'image par le procédé d'Huffman pour aboutir à notre image compressée. [Figure 24 : codage RLE \(wikipedia.fr\)](http://wikipedia.fr)

2.3.2 Application à la problématique

Le format JPEG étant une compression de l'image initiale, il n'est pas approprié pour l'étude de la teinte des pixels. Pour cela il est préférable de travailler sur une image au format BITMAP. Mais sachant que ce format non compressé est beaucoup plus lourd aussi, lors de l'affichage de l'ortho-image dans SolidWorks, le JPEG est plus adéquat car il y a un gain de temps lors de l'ouverture du fichier.

Exemple dans le logiciel, le temps d'ouverture d'une même image en JPEG est quasiment instantané alors que pour le BITMAP nécessite une dizaine de secondes.

2.4 Le processus d'import d'ortho-images

Afin de pouvoir afficher les ortho-images dans SolidWorks il a fallu faire un enchainement de plusieurs étapes. Avant la création de ce plug-in chaque phase était faite « à la main » par l'utilisateur et cela prenait du temps et demandait beaucoup de manipulation. Aujourd'hui l'utilisateur n'a qu'à choisir l'ortho-image qu'il souhaite ajouter à son projet et le plug-in gère son importation. Bien évidemment ce processus est directement lié au logiciel SolidWorks et ne conviendrait pas à d'autres logiciels de dessin 3D.

Les étapes d'import d'ortho-images sont les suivantes :

- La lecture du fichier texte contenant les informations des coordonnées des quatre coins de l'image.
- Ouvrir une esquisse 3D et créer aux moins trois points qui correspondent aux coins de l'image.
- Sélectionner ces points pour créer un plan.
- Convertir les points 3D en 2D dans une esquisse.

Explication: SolidWorks est un logiciel qui fonctionne avec différents systèmes de coordonnées. Il existe un système de coordonnées global mais pour chaque esquisse 2D créée le logiciel projette l'origine du système global sur le plan d'esquisse et définit les axes de façon à ce qu'ils soient colinéaires avec les « côtés » du plan.

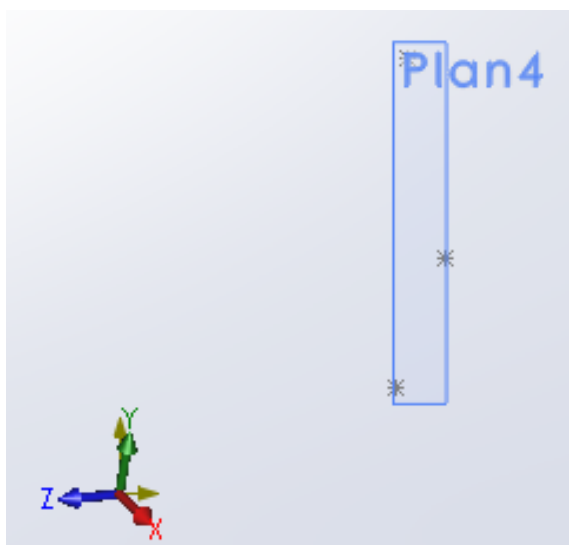


Figure 26 : Vue normale au plan

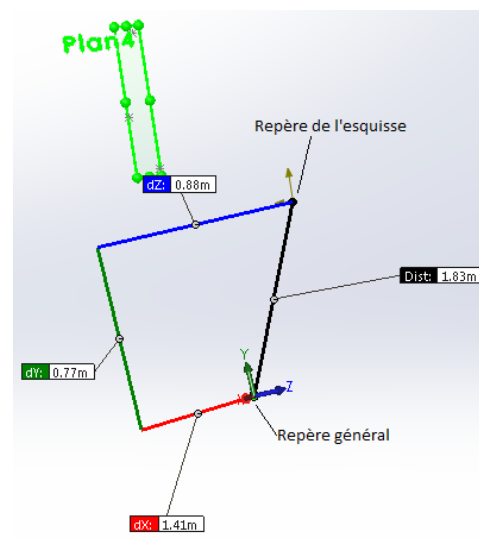


Figure 25 : Origine de l'esquisse

Sur la figure 26, la vue est suivant la normale au plan 4. Les deux repères ont leur origine « confondue » sur la vue. Le repère de l'esquisse est contenu dans le plan d'esquisse comme représenté sur la figure 25. Pour définir les axes, SolidWorks crée un plan en définissant un rectangle englobant les trois points le plus petit possible. La direction de la longueur et la largeur détermine les axes du repère d'esquisse en 2D.

Une fois les points 3D convertis en point2D dans l'esquisse, il est possible de récupérer leurs coordonnées 2D dans l'esquisse.

- Créer une esquisse sur le plan, dessiner un rectangle avec ces points 2D puis convertir ce rectangle en surface de référence.
- L'ultime étape consiste à sélectionner cette surface et lui appliquer un calque, notre ortho-image. Il est nécessaire une propriété du décalque : un possible angle de rotation car par défaut le décalque se place orthogonalement sur la surface.

Voici les différents éléments créés par ce processus dans l'arbre de Solidworks (figure27) :

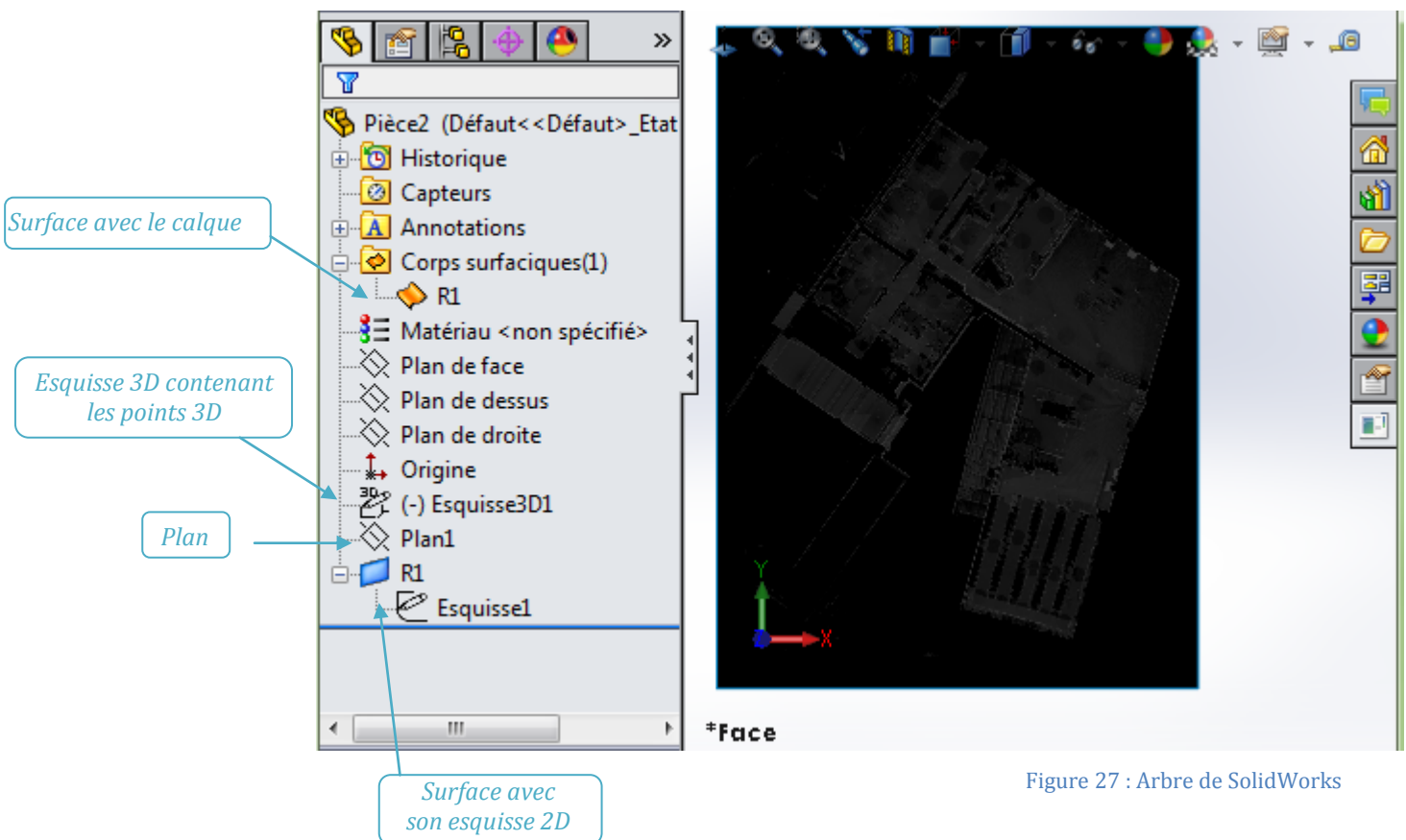


Figure 27 : Arbre de SolidWorks

2.5 Comment gérer et sauvegarder les ortho-images dans le plug-in

2.5.1 Gérer les ortho-images

Une fois les ortho-images ajoutées dans SolidWorks, le plug-in doit offrir des fonctions pour les gérer. Tout d'abord à chaque insertion, les ortho-images s'ajoutent petit à petit dans une liste.

Cette liste est ce qu'on appelle en langage informatique une CheckBoxList. En effet chaque petit carré se trouvant avant le nom de l'ortho-image est une CheckBox (une boîte qui peut être validée ou non). Lorsque la case est validée, l'ortho-image est visible dans l'espace de dessin de SolidWorks. A contrario, lorsqu'elle est décochée, l'ortho-image est non visible. Cette propriété joue sur la surface créée dans le dossier « Corps surfaciques » qui a la possibilité d'être caché ou visible.

Comme vu précédemment, les ortho-images peuvent être ajoutées individuellement, par plusieurs ou directement en sélectionnant un dossier. De la même façon il est possible de supprimer les ortho-images en les sélectionnant dans la liste.

2.5.2 Sauvegarde des ortho-images

Lors de l'utilisation du plug-in il est nécessaire que les ortho-images soient sauvegardées dans le projet SolidWorks pour être présentes lors d'une prochaine ouverture. Pour cela SolidWorks offre la possibilité lors de l'enregistrement du projet d'insérer un fichier texte au format .SLDPRT propre au logiciel.

Pour obtenir ce fichier texte contenant les ortho-images à sauvegarder un premier traitement doit être effectué. En effet pour chaque ortho-image il est obligatoire d'enregistrer toutes ces propriétés pour être réutilisées par la suite. Les propriétés à enregistrer sont :

- le nom de l'ortho-image,
- le chemin d'accès au fichier texte contenant les coordonnées de l'image,
- le chemin d'accès au fichier image au format JPEG et BITMAP pour les « images solides »,
- le type : est-ce qu'il s'agit d'une ortho-image simple ou d'une « image solide ».

Pour stocker les ortho-images avec leurs propriétés, nous allons utiliser le langage XML.

2.5.2.1 Qu'est-ce que le XML

Les initiales XML correspondent à eXtensible Markup Language ce qui signifie en français : un langage de balisage extensible.

Le langage, cela indique que ce type de fichier est créé pour donner des informations. Pour cela c'est un moyen de communication et d'échange entre les applications informatiques.

Un langage de balisage : cela indique que les données sont qualifiées par des « étiquettes » qui expliquent le contenu. Prenons un exemple simple pour illustrer : dans Charles Baudelaire, « Charles » prendra l'étiquette prénom alors que « Baudelaire » sera un nom de famille.

Un langage extensible : cela indique deux choses :

- en fonction du besoin, il est possible d'utiliser autant d'étiquette que nécessaire,
- avec des étiquettes définies, il est toujours possible d'en rajouter même si ce n'est plus la même personne qui utilise le fichier.

2.5.2.2 Pourquoi utiliser le XML ?

Le XML est avant tout un fichier texte mais qui possède comme décrit précédemment une structure particulière. L'avantage de ce langage est qu'il est facilement compréhensible par des personnes non-initiées au langage informatique. A l'aide des balises, un système informatique est aussi apte d'en lire le contenu. C'est donc tout l'intérêt de ce langage de pouvoir être lu à la fois par un humain et un système informatique.

De plus, il existe de nombreux outils informatiques liés au XML qui permettent de manipuler les documents XML : XSD, XSLT, XQuery,...

2.5.2.3 Exemple de fichier XML

Prenons l'exemple de célèbres scientifiques :

```
<?xml version="1.0" encoding="UTF-8"?>
<savants>
  <!-- Quelques physiciens importants -->
  <savant id="phys0">
    <identité prénom="Galileo" nom="Galilei"/>
    <dates naissance="1564-02-15" décès="1642-01-08"/>
    <contributions>Relativité du mouvement et système
    héliocentrique</contributions>
  </savant>
  <savant id="phys1">
    <identité prénom="Isaac" nom="Newton"/>
    <dates naissance="1643-01-04" décès="1727-03-31"/>
    <contributions>Gravitation universelle, principe d'inertie,
    décomposition de la lumière & amp; calcul infinitésimal
    </contributions>
  </savant>
  <savant id="phys2">
    <identité nom="Einstein" prénom="Albert"/>
    <dates naissance="1879-03-14" décès="1955-04-18"/>
    <contributions>Relativités restreinte et générale, nature
    corpusculaire de la lumière et effet photoélectrique
    </contributions>
  </savant>
</savants>
```

Figure 28 : Exemple XML

Rentrons un peu dans les détails (figure 28) :

1^{ère} ligne : < ?xml version= »1.0 « encoding= »UTF-8 « ?>

C'est ce qu'on appelle le prologue, il désigne la version qui sert de base à l'écriture du fichier puis l'encodage de caractère utilisé. Cette ligne n'est pas obligatoire mais permet de clarifier les paramètres lorsque l'utilisateur n'est pas toujours la même personne.

Les balises : ouvrantes <savants> ou fermantes </savants> ou les auto-fermantes telles que <identité .../> ou <dates.../>. Chaque élément est défini par des balises. A l'intérieur de balise ouvrante et fermante d'un élément non nul peuvent se trouver d'autres éléments. En effet nous retrouvons ce phénomène avec savant. Il existe donc une relation hiérarchique entre les éléments. Un fichier XML est structuré comme une arborescence.

Un élément joue un rôle important dans un fichier XML est l'élément racine. Celui-ci est obligatoire unique et il doit englober tous les autres éléments. Ici cet élément est « savants ».

Afin de définir ce que contient l'élément, on utilise des attributs. Pour l'élément identité, il est précisé l'attribut prénom et nom. Chaque attribut a un nom et une valeur. Par exemple l'attribut prénom du dernier scientifique a pour valeur Albert.

La 3^{ème} ligne est une ligne de commentaire : tout ce qui est contenu entre les balises `<!-- -et -->` apporte des informations, des compléments mais n'est pas une donnée du document. En programmation les commentaires sont très importants car ils facilitent la lecture d'un code, surtout si plusieurs développeurs travaillent sur le même projet.

Il existe donc des règles d'écriture pour un fichier XML :

- Il y a une différenciation entre les majuscules et les minuscules, un élément NOM est différent de nom.
- Un élément ne peut pas avoir un nom qui commence par un chiffre.
- Tous les éléments doivent être obligatoirement fermés avec un ordre à respecter : `<élément1><élément2> ... </élément1></élément2>` n'est pas possible, il faut respecter l'ordre d'ouverture : `<élément1><élément2> ... </élément2></élément1>`.
- Chaque attribut a sa valeur entre guillemets.
- L'élément racine est unique.

2.5.2.4 Application du XML au cas des ortho-images

Lors de chaque modification de la liste des ortho-images, ajout ou suppression, le fichier XML est mis à jour. Suivant les règles et l'exemple précédent ce fichier est composé de (figure29) :

```
<?xml version="1.0" encoding="UTF-8"?>
<Liste_orthoimages>
  <!-- Toutes les ortho-images contenues dans la liste du plug-in-->
  <ortho id = "o1">
    <identification nom = "test1"/>
    <chemin_accès texte = "C:\Users\GEOSAT_3D\Desktop\test1.txt"
    imageJPEG = "C:\Users\GEOSAT_3D\Desktop\test1.jpg"
    imageBITMAP=""/>
    <type kind = "2D" />
  </ortho>
  <ortho id = "o2">
    <identification nom = "test2"/>
    <chemin_accès texte = "C:\Users\GEOSAT_3D\Desktop\test2.txt"
    imageJPEG = "C:\Users\GEOSAT_3D\Desktop\test2.jpg"
    imageBITMAP="C:\Users\GEOSAT_3D\Desktop\test2.bmp"/>
    <type kind = "3D" />
  </ortho>
```

Figure 29 : XML propre aux ortho-images

Dans cet exemple la première ortho-image de la liste est nommée « test1 » et c'est une ortho-image « simple », elle ne permet pas de dessiner en 3D. A contrario la seconde est une « image solide » et le chemin d'accès au fichier image BITMAP est donc nécessaire pour avoir les renseignements sur la valeur des pixels.

Une fois le fichier XML à jour lors de l'enregistrement du projet, il est converti en fichier texte afin de pouvoir être ajouté au fichier propre de SolidWorks. Lors de la réouverture du fichier la conversion inverse sera nécessaire, le fichier texte sera transformé en fichier XML afin de pouvoir récupérer la liste d'ortho-images avec toutes leurs propriétés.

2.6 Exemple d'utilisation



Figure 30 : cité administrative de Bordeaux (pss-archi.eu)

Suite à une commande faite par la préfecture de Gironde pour réaliser une modélisation 3D de la cité administrative de Bordeaux, le plug-in a été sollicité pour importer les ortho-images et voici le résultat (voir annexe 2)

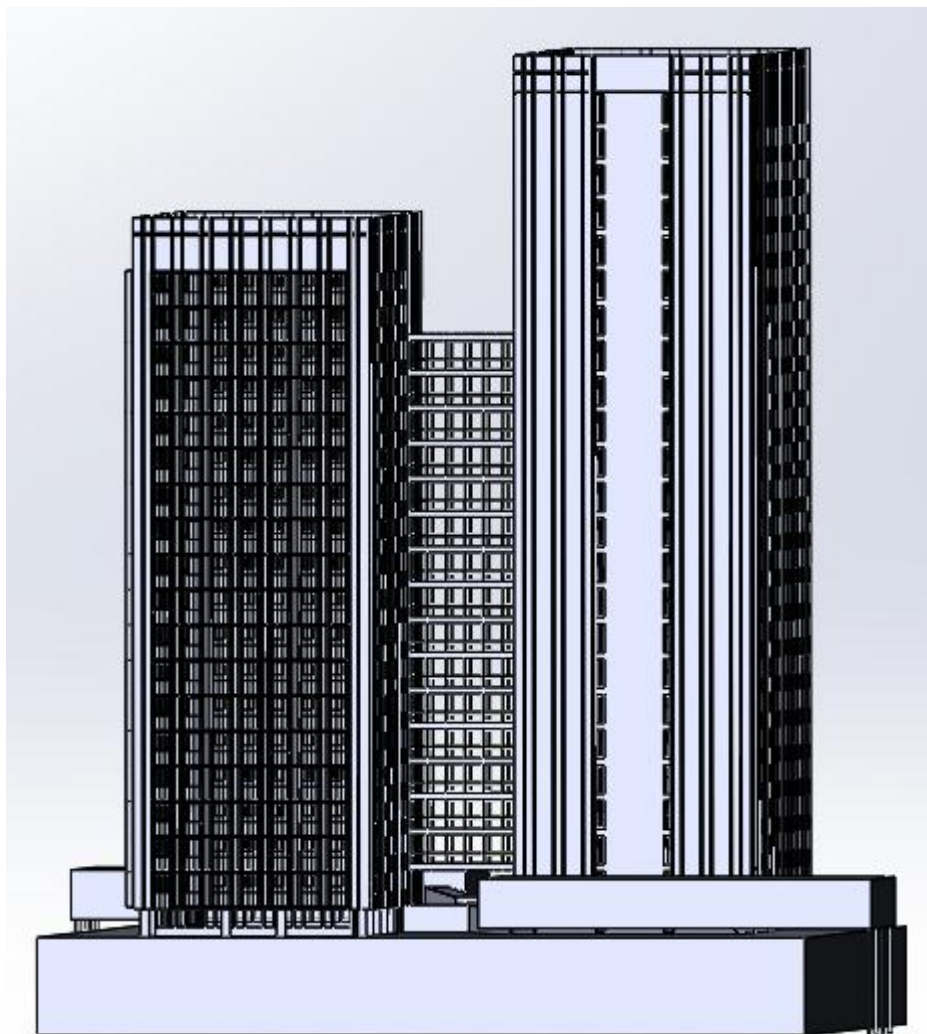


Figure 31 : résultat obtenu avec SolidWorks

Chapitre 3 : Manipulation d'une image solide générée par R3

Grâce aux images solides générées par R3, plusieurs fonctionnalités ont été développées pour automatiser la modélisation en 3D.

3.1 Récupération de l'information d'altitude dans l'image

La propriété principale d'une image solide est de contenir l'information d'altitude dans chacun de ses pixels. Les différentes étapes permettant de récupérer cette information sont les suivantes :



3.1.1 Changement de système de coordonnées

Comme expliqué dans la partie II, les ortho-images sont ajoutées à l'aide de la première fonctionnalité du plug-in. Afin de pouvoir récupérer la valeur d'altitude de chaque pixel, il est nécessaire de se placer dans le système de coordonnées de l'image. L'origine étant le coin en haut à gauche. Il est donc indispensable de procéder à un calcul de changement de repère avec une combinaison de translation et de rotation.

3.1.1.1 La translation : le changement d'origine

Considérons le point P dont les coordonnées sont [E,N] dans le repère d'origine O en 2D (figure33) :

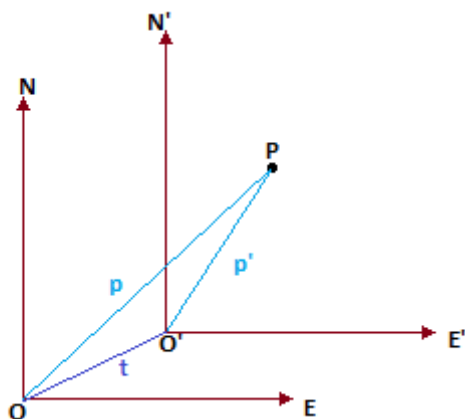


Figure 33 : translation

Pour trouver la valeur de la translation t :

$$p' - p = \begin{bmatrix} E' \\ N' \end{bmatrix} - \begin{bmatrix} E \\ N \end{bmatrix} = t$$

$$t = p' - p = O - O'$$

3.1.1.2 Les effets de la rotation des axes (figure 32)

$$p' = \begin{bmatrix} E' \\ N' \end{bmatrix} = \begin{bmatrix} \cos r & \sin r \\ -\sin r & \cos r \end{bmatrix} \begin{bmatrix} E \\ N \end{bmatrix}$$

$$p = \begin{bmatrix} E \\ N \end{bmatrix} = \begin{bmatrix} \cos r & -\sin r \\ \sin r & \cos r \end{bmatrix} p'$$

$$p' = Rp \text{ et } p = R^T p'$$

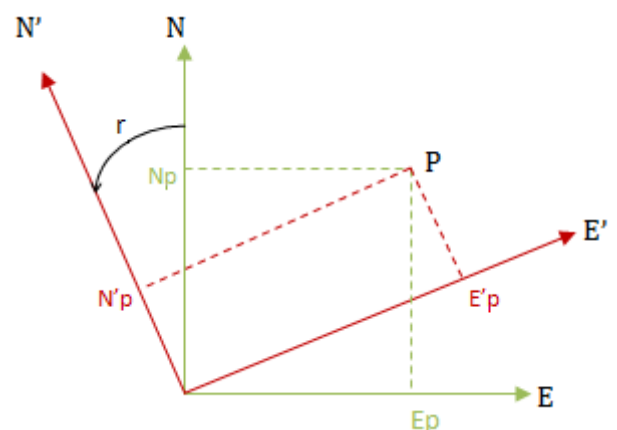


Figure 32 : rotation

3.1.1.3 Combinaison de la translation et la rotation

La combinaison de la translation et de la rotation se traduit par :

$$p' = t + Rp$$

$$p = R^T (p' - t) = R^T p' - R^T t = R^T p' + t'$$

Enfin, un facteur d'échelle peut être introduit, c'est-à-dire un rapport liant l'unité de longueur des deux systèmes. Considérons le facteur λ , cela devient :

$$p' = t + \lambda Rp$$

$$p = \lambda^{-1} R^T (p' - t) = \lambda^{-1} R^T p' + t'$$

3.1.1.4 Changement de repère en 3D

La translation est décomposée en trois « sous » translations comme ceci : $t = \begin{bmatrix} tx \\ ty \\ tz \end{bmatrix}$.

La rotation est affectée par la même décomposition, trois « sous » rotation suivant les axes X, Y et Z et les trois angles de rotations rx, ry, rz . Ces trois « sous » rotation donnent les matrices suivantes :

$$R_x(rx) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos rx & \sin rx \\ 0 & -\sin rx & \cos rx \end{bmatrix}$$

$$R_y(ry) = \begin{bmatrix} \cos ry & 0 & -\sin ry \\ 0 & 1 & 0 \\ \sin ry & 0 & \cos ry \end{bmatrix}$$

$$R_z(rz) = \begin{bmatrix} \cos rz & \sin rz & 0 \\ -\sin rz & \cos rz & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R(rx, ry, rz) = R_z(rz)R_y(ry)R_x(rx)$$

La matrice de rotation finale est la suivante :

R =

$$\begin{bmatrix} \cos(ry) \cos(rz) & \cos(ry) \sin(rz) + \sin(rx) \sin(ry) \cos(rz) & \sin(rx) \sin(rz) - \cos(rx) \sin(ry) \cos(rz) \\ -\cos(ry) \sin(rz) & \cos(rx) \cos(rz) - \sin(rx) \sin(ry) \sin(rz) & \sin(rx) \cos(rz) + \cos(rx) \sin(ry) \sin(rz) \\ \sin(ry) & -\sin(rx) \cos(ry) & \cos(rx) \cos(ry) \end{bmatrix}$$

La transformation totale dépend de sept paramètres : trois translations, trois rotations et un facteur d'échelle. Cette transformation se nomme : transformation d'Helmert.

Les rotations et translations étant exprimées ci-dessus il reste à déterminer le facteur d'échelle. Il correspond à la valeur de la résolution de l'image BITMAP sur laquelle on souhaite récupérer les informations sur les pixels. La résolution d'une image est la valeur du rapport entre le nombre de pixel qui compose l'image et ces dimensions.

Par exemple si l'image mesure 65m x 95m pour 13000pixels x 19000pixels alors sa résolution sera de : 0.005m. $\left(\frac{65}{13000} = \frac{95}{19000} = 0.005\right)$

Pour le passage du système général au système propre à l'image il faut donc diviser les coordonnées obtenues après la translation et la rotation par la résolution.

3.1.2 Algorithme de Bresenham

3.1.2.1 Présentation

Une fois placé dans le système de coordonnées « image », nous pouvons récupérer l'altitude de chaque pixel. Avant cela n'oublions pas que nous cherchons à modéliser une droite en 3D. Pour avoir la troisième dimension, il est essentiel de « sélectionner » tous les pixels qui se trouvent sur cette droite. Un algorithme d'infographie permet de réaliser cela : l'algorithme de Bresenham. Il est une des bases de l'infographie. Comme l'indique son nom, il a été développé par Jack E. Bresenham en 1962 durant ces années de recherche en laboratoire informatique chez IBM.

Le principe consiste à passer un segment défini de manière continue à un segment défini de manière discrète avec des pixels. Pour cela il est nécessaire de respecter trois impératifs :

- Chaque pixel du segment discret devra être traversé par le segment continu.
- Chaque pixel du segment discret doit être touché par un autre pixel soit par un côté (pixels orange), soit par un de ses sommets (pixel bleus).

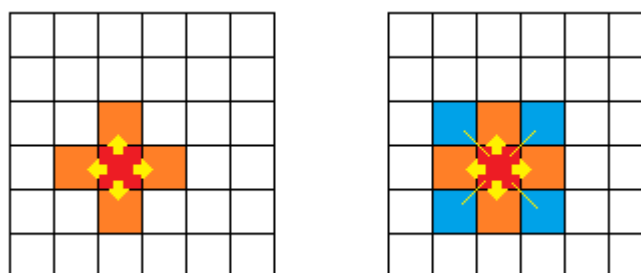


Figure 34 : pixels voisins

- Il faut un nombre minimum de pixels sélectionnés.

Voici quelques exemples de bons et mauvais tracés de pixel :

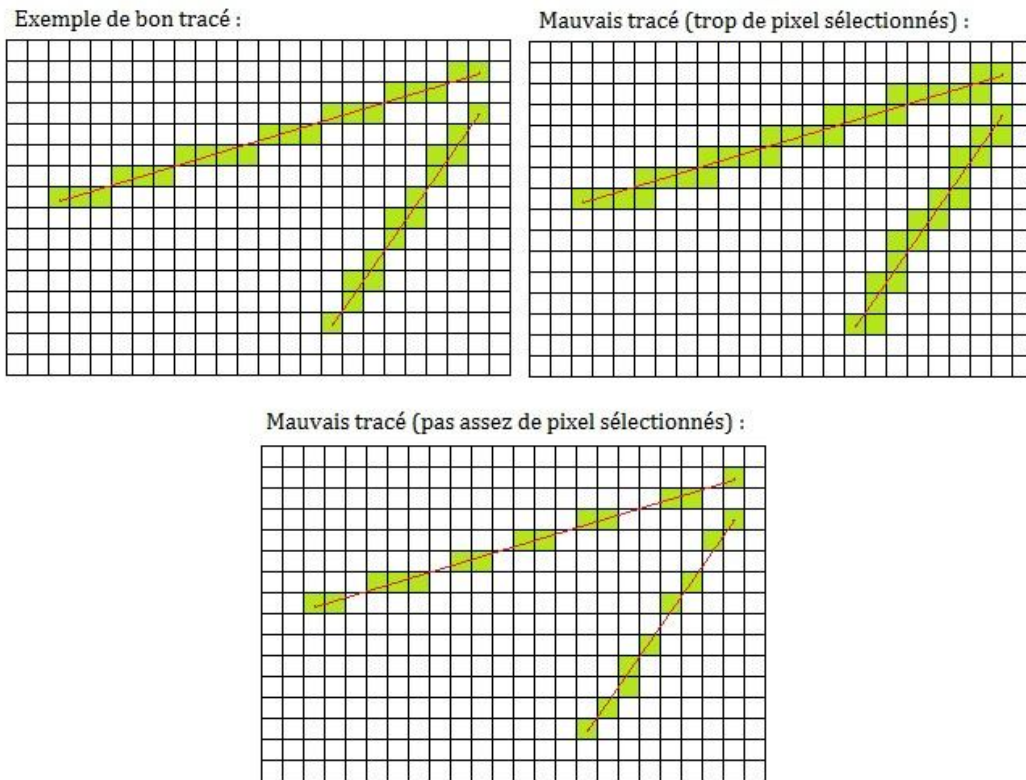


Figure 35 : exemples tracés

Pour obtenir un bon tracé l'objectif est le suivant :

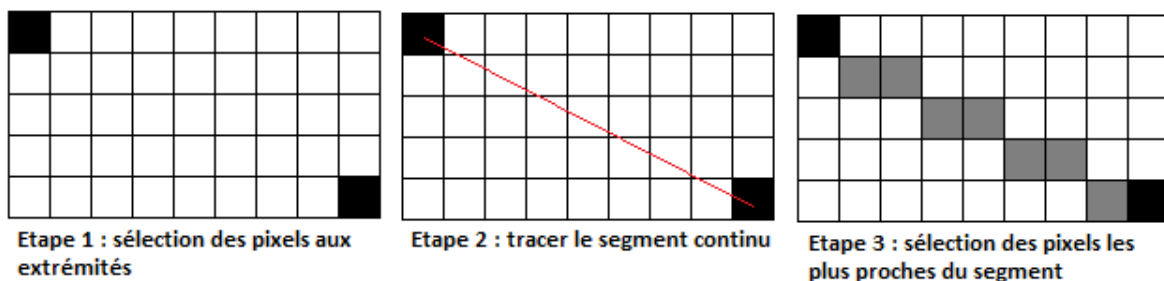
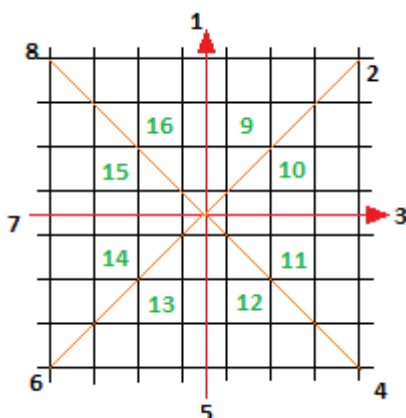


Figure 36 : procédure de sélection des pixels

3.1.2.2 Fonctionnement de l'algorithme

Afin de déterminer quels pixels doivent être sélectionnés, l'algorithme procède de la manière suivante (voir annexe 3 pour l'algorithme) :



- Division en plusieurs cas (figure 37): il s'agit de découper le maillage de pixels en huit parties grâce à des lignes horizontales, verticales et les diagonales qui se coupent au centre du pixel de départ. Ces droites définissent donc seize régions qui entourent le premier pixel. Le pixel d'arrivée est situé dans une de ces régions comme illustré figure.

- 1^{er} cas : le pixel d'arrivée se trouve sur les demi-droites numérotées de 1 à 8.

S'il est sur les droites verticales ou horizontales, il faudra sélectionner les pixels qui se touchent successivement par leur côté en direction du point d'arrivée. S'il est une diagonale alors ce sera les pixels qui se touchent par leur sommet.

- 2nd cas : le pixel d'arrivée se trouve dans l'une des régions numérotées de 9 à 16.

Prenons l'exemple avec un pixel se trouvant dans la région 10. Considérons un pixel P de coordonnées (x,y) , les deux pixels suivants potentiellement sélectionnés sont $P1(x+1,y)$ ou $P2(x+1,y+1)$.

Sur le schéma suivant (*figure 38*) sont représentés les pixels P, P1 et P2. Les centres de ces pixels sont reliés par le trait de couleur bleu. Le trait rouge est le segment qu'on souhaite modéliser discrètement.

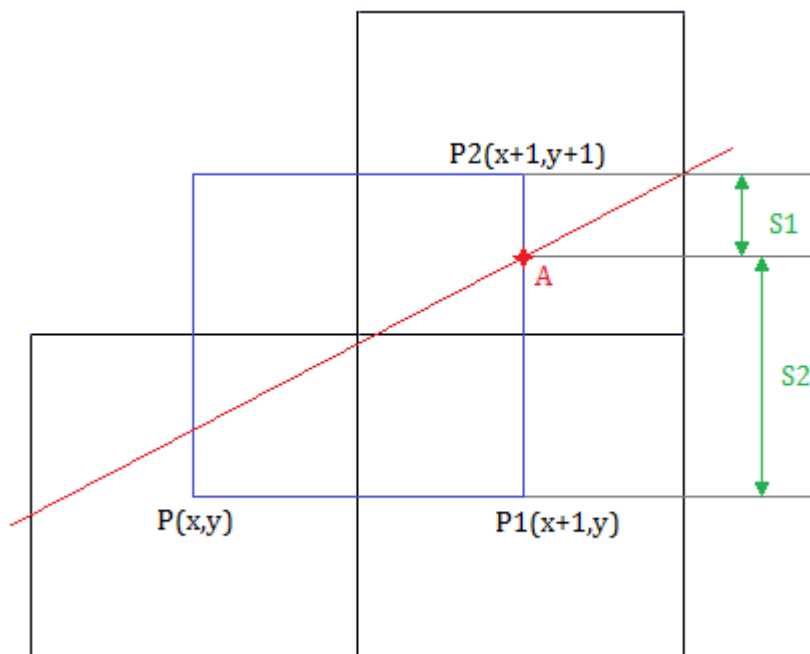


Figure 38 : Sélection du pixel

Soit d , la pente de la droite passant par le point de départ $D1(x1, y1)$ et le point d'arrivée $D2(x2,y2)$.

$$d = \frac{y2 - y1}{x2 - x1}$$

Le choix de sélectionner le pixel P1 ou P2 se fait en fonction de la position du point A par rapport au milieu du segment $[P1P2]$. Les coordonnées de A sont $(x + 1, d \times (x + 1))$.

- Si $S2 > S1$ (comme sur l'exemple), le pixel sélectionné sera le pixel P2
- Si $S2 < S1$ alors on sélectionne le pixel P1.

Le processus se répète ainsi jusqu'à arriver au pixel d'extrémité du segment.

3.1.3 Récupération de l'altitude

Pour chaque pixel sélectionné, nous avons ses coordonnées il est donc possible de récupérer la valeur de la teinte. Comme expliqué précédemment à chaque teinte correspond une altitude.

Cette valeur sera stockée durant les prochaines étapes et sera utilisée lors du dessin en 3D.

3.1.4 Projeté orthogonal du pixel

Une fois le chemin de pixel établi avec chacun une altitude, la prochaine étape pour aboutir à une représentation en 3D est de calculer le projeté orthogonal du pixel sur la droite. En effet, le but étant de modéliser une droite si nous gardons le chemin de pixel tel que nous l'avons obtenu dans la partie précédente nous obtiendrons un « escalier » (figure 39) et non pas quelque chose de rectiligne.

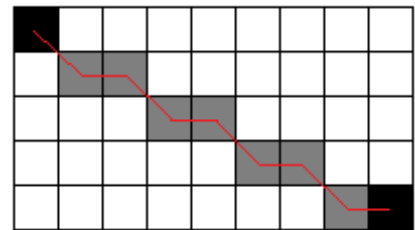


Figure 39 : chemin de pixel

3.1.4.1 Equation d'une droite

Considérons une droite d et un point M à projeter sur cette droite (figure 40):

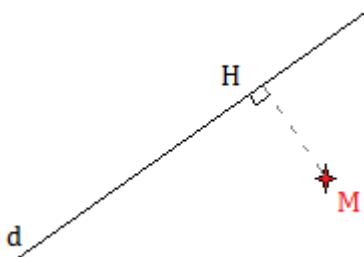


Figure 40 : projeté orthogonal

d a pour équation : $ax + by + c = 0$

Les extrémités du segment sont connues en coordonnées $A[x_1, y_1]$ et $B[x_2, y_2]$. Il est donc possible de calculer la valeur des coefficients a , b et c .

$$a = y_2 - y_1$$

$$b = x_1 - x_2$$

$$c = -(b \times y_1 + a \times x_1)$$

3.1.4.2 Projeté orthogonal

Une fois l'équation de la droite connue, le projeté orthogonal peut être calculé :

\vec{u} étant le vecteur unitaire de la droite d alors : $\vec{u} \cdot \overrightarrow{HM} = 0$ et $HM = \begin{bmatrix} x_M - x_H \\ y_M - y_H \end{bmatrix}$.

D'où $-b(x_M - x_H) + a(y_M - y_H) = 0$.

Équation 1 : $bx_H - ay_H - bx_M + ay_M = 0$

Équation 2 : $ax_H + by_H + c = 0$

Après résolution du système d'équations les coordonnées du point H sont :

$$x_H = \frac{b^2 x_M - a b y_M - a c}{a^2 + b^2}$$

$$y_H = \frac{-a b x_M + a^2 y_M - b c}{a^2 + b^2}$$

Chaque pixel peut ainsi être projeté sur la droite d'origine. (figure41)

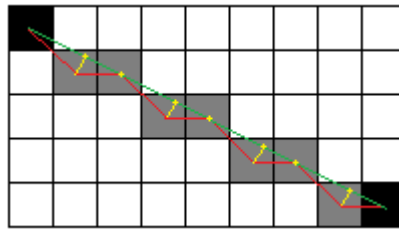


Figure 41 : chemin projeté sur la ligne

3.1.5 Retour dans le système de coordonnées général de SolidWorks

Afin de pouvoir dessiner en trois dimensions dans SolidWorks il est indispensable d'avoir les coordonnées de chaque pixel projeté sur la droite dans le système général et non dans le système de l'image.

Pour cela il suffit d'appliquer la transformation inverse de celle faite précédemment.

$$p' = t + Rp$$

$$p = R^T (p' - t) = R^T p' - R^T t = R^T p' + t'$$

La translation devient donc : $t' = -t$

Quant à la matrice de rotation c'est la transposée de la précédente : $R2 = R^T$

$$R2 = \begin{bmatrix} \cos(ry) \cos(rz) & -\cos(ry) \sin(rz) & \sin(ry) \\ \cos(ry) \sin(rz) + \sin(rx) \sin(ry) \cos(rz) & \cos(rx) \cos(rz) - \sin(rx) \sin(ry) \sin(rz) & -\sin(rx) \cos(ry) \\ \sin(rx) \sin(rz) - \cos(rx) \sin(ry) \cos(rz) & \sin(rx) \cos(rz) + \cos(rx) \sin(ry) \sin(rz) & \cos(rx) \cos(ry) \end{bmatrix}$$

3.1.6 Dessin en 3D

Tous les éléments nécessaires pour le dessin en trois dimensions sont calculés dans les étapes précédentes :

- la position de chaque pixel dans le système général de SolidWorks
- la valeur de la hauteur à appliquer pour chaque pixel.

Ainsi il est possible de pouvoir dessiner en trois dimensions.

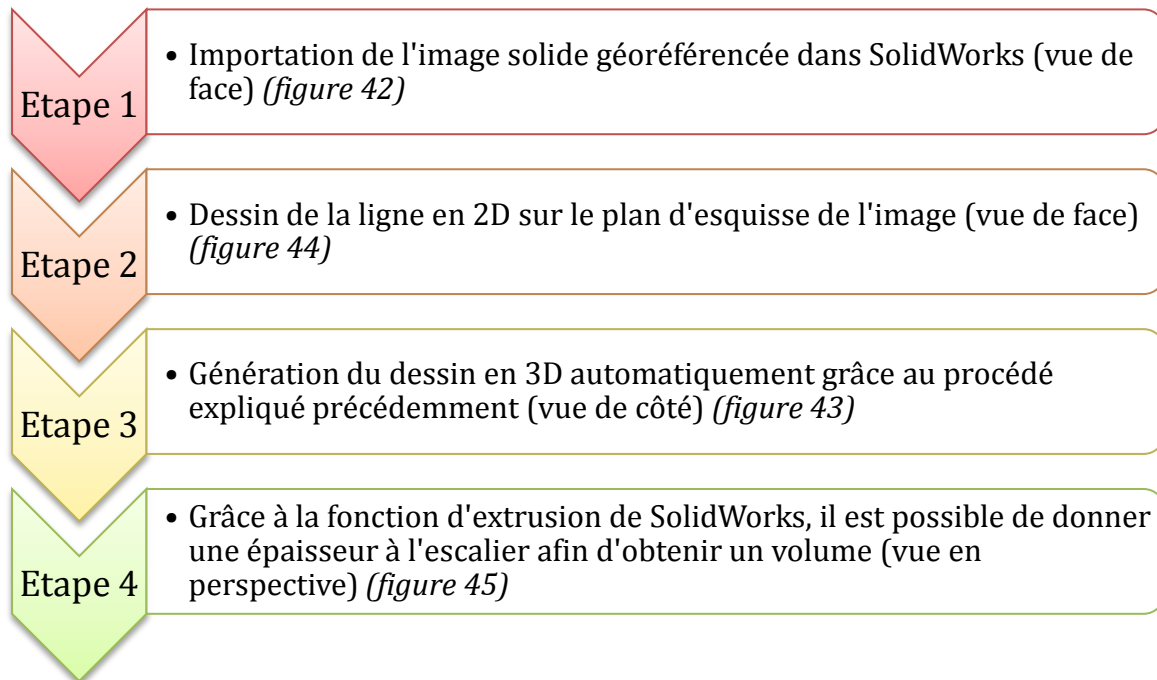
3.1.6.1 L'ortho-image dans SolidWorks

Comme expliqué dans la première partie, il est préférable de travailler avec une ortho-image dont l'altitude est représentée par une nuance de teinte pour une meilleure visualisation et une meilleure précision. Pour des raisons de développement du logiciel R3 par Gexcel il est pour l'instant possible de travailler qu'avec des ortho-images dont la variation d'altitude est symbolisée par des nuances de gris.

3.1.6.2 Exemple d'utilisation

Voici un exemple de dessin en trois dimensions d'un escalier. Afin de ne pas confondre l'altitude de chaque pixel représenté par les nuances de gris et le fond de l'image, ce dernier est en rose sur l'image solide.

Plus les marches sont hautes plus le gris devient clair.



Etape 1 :

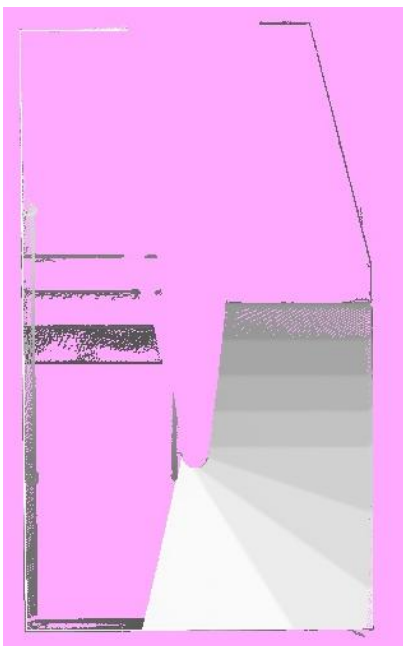


Figure 42 : image solide

Etape 2 :



Figure 44 : droite en 2D

Etape 3 :

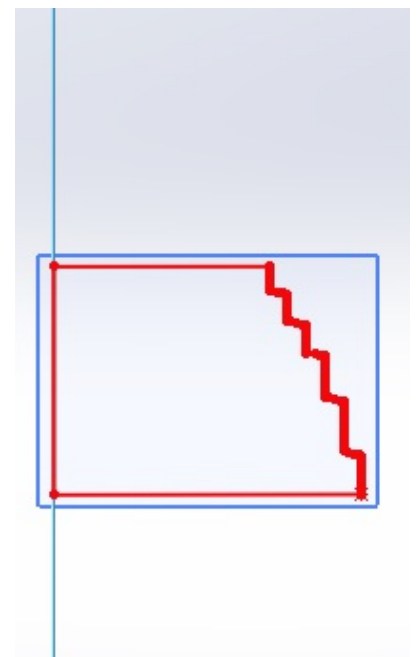


Figure 43 : droite en 3D (vue de côté)

Etape 4 :

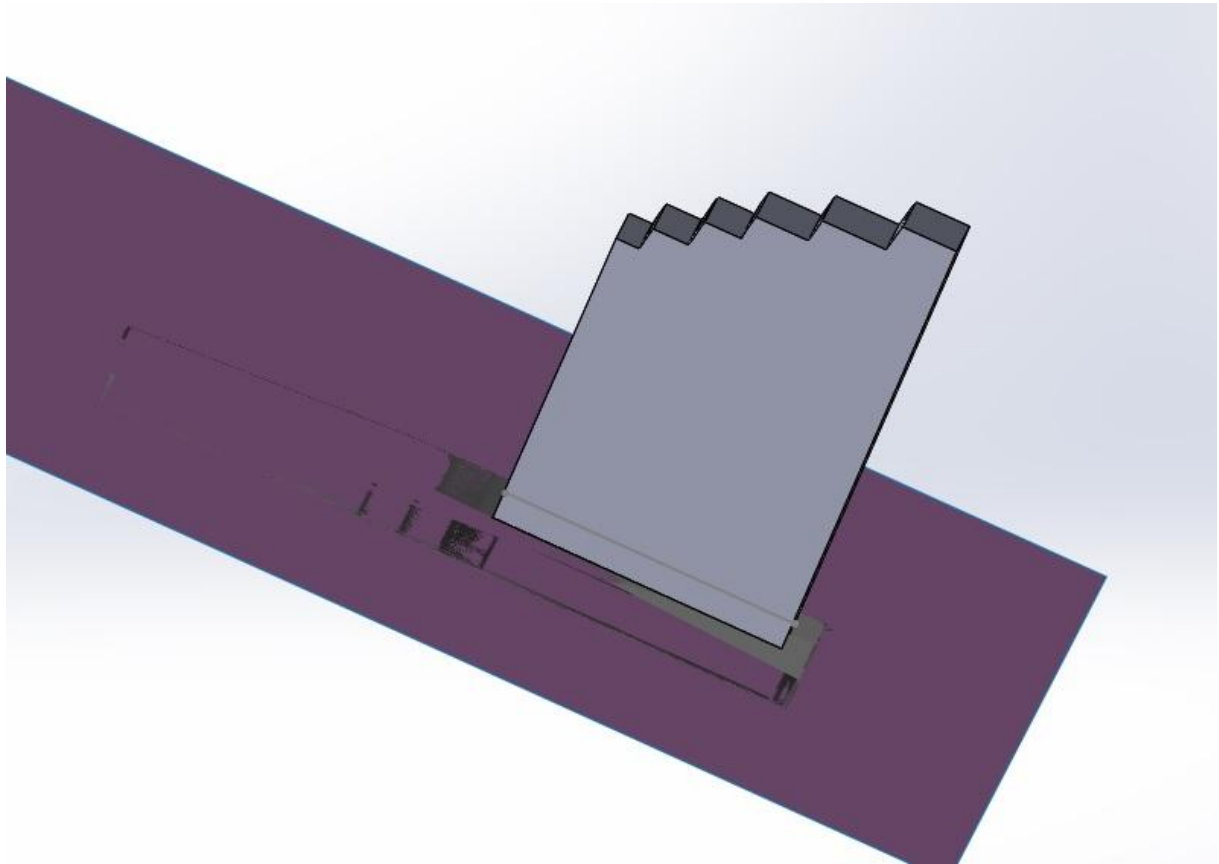


Figure 45 : volume en 3D de l'escalier

3.2 Création d'un MNT

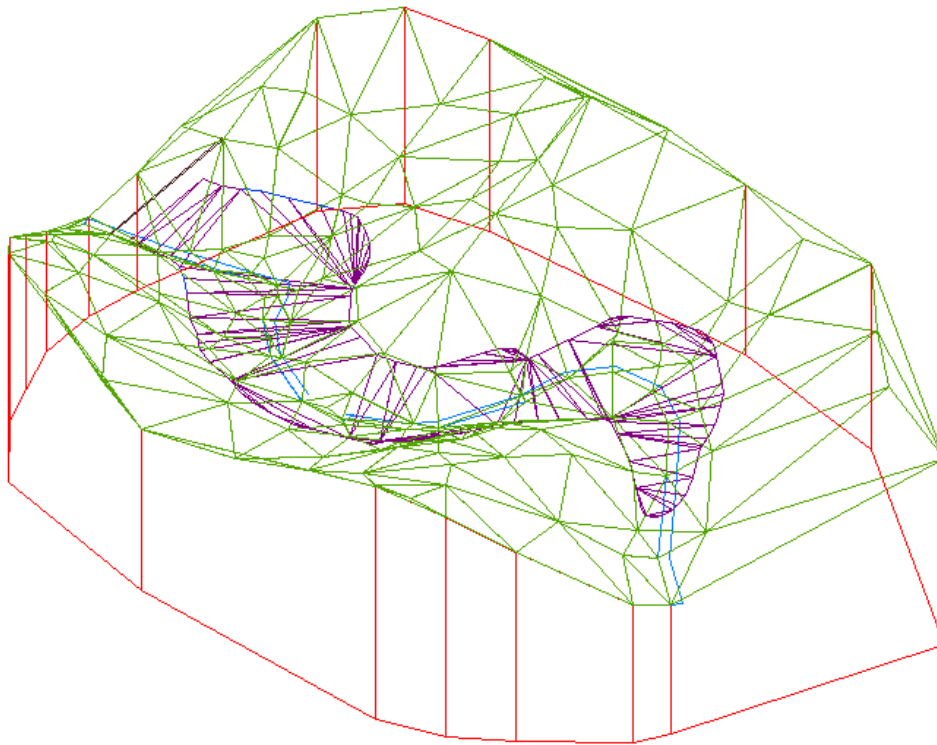


Figure 46 : Exemple MNT (caillie13.btstopo.free.fr)

3.2.1 Qu'est-ce qu'un MNT ?

MNT signifie Modèle Numérique de Terrain. C'est une représentation du relief terrestre avec une forme mathématique simplifiée. Il existe différents types de MNT, ils sont différenciés selon le maillage utilisé (rectangulaire, triangulaire, hexagonal,...) (figure 46).

L'utilisation d'un MNT se retrouve dans de nombreuses applications :

- Réalisation de cartes de relief avec des profils topographiques,
- Visualisation 3D,
- Positionnement GNSS,
- Utilisation dans les SIG (Systèmes d'information géographique),...

L'IGN met à disposition dans l'espace professionnel de son site des MNT à différentes échelles. Ceux-ci sont obtenus par interpolation à partir de courbes de niveau et de points cotés. Exemple avec la BD ALTI® qui permet de représenter et analyser le territoire à moyenne échelle. Créée entre 1987 et 1995, la BD ALTI provient de la numérisation de cartes et de la restitution photogrammétrique.

Différence entre MNT, MNE et MNS (figure 47)

MNT = modèle numérique de terrain, il représente le terrain.

MNE = modèle numérique d'élévation, représentation de tout ce qui se trouve au-dessus du sol.

MNS = modèle numérique de surface, décrit toute la surface (sol + sur-sol).

$$MNS = MNT + MNE$$

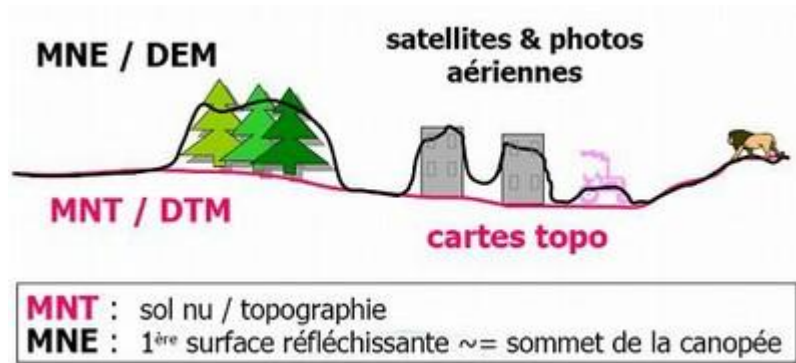


Figure 47 : MNT, MNS et MNE (ensg.eu)

Contrairement au MNT, le MNE prend en compte le relief mais aussi le sur-sol : élévation des bâtiments, des arbres (la canopée). Lors de la capture d'images aériennes ou satellites, c'est un MNE et il est nécessaire de recourir à des traitements d'images pour pouvoir en extraire un MNT.

3.2.2 De l'image solide à un MNT

Comme expliqué précédemment, il est possible de récupérer la valeur de l'altitude d'un point sur une image solide. Sur le même procédé que lors du dessin de la ligne en 3D. L'utilisateur peut choisir des points sur l'image (figure 50), ils seront ensuite élevés de leur altitude et par un procédé de triangulation voici le résultat obtenu (figures 49 et 48) :

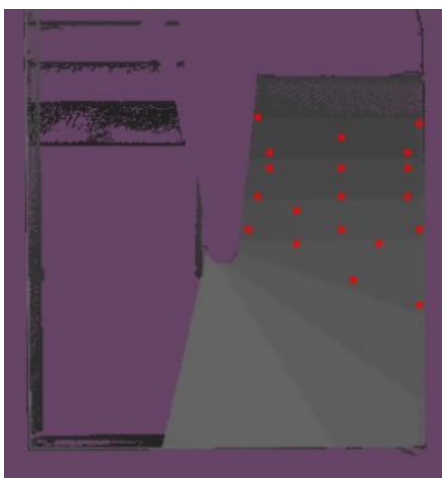


Figure 50 : points du MNT

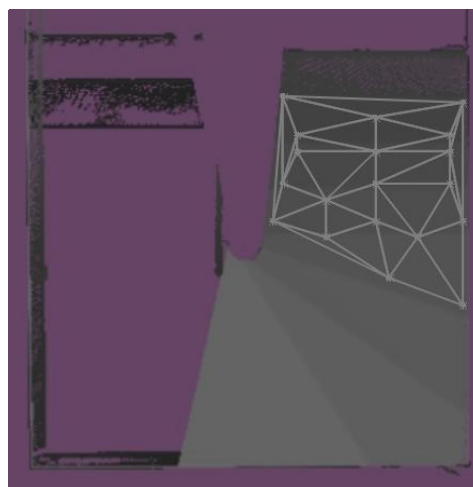


Figure 49 : MNT (vue de face)

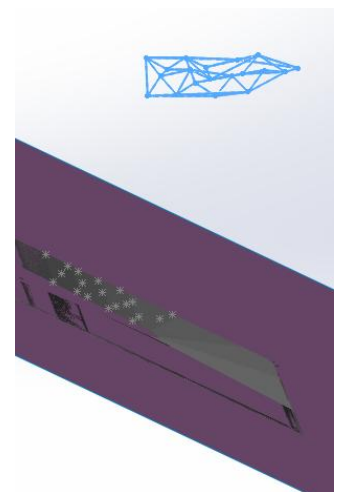


Figure 48 : MNT (vue perspective)

Pour arriver à la création de ce MNT plusieurs procédés mathématiques ont été nécessaires dont la triangulation de Delaunay.

3.2.3 La triangulation de Delaunay

La triangulation de Delaunay est l'une des méthodes les plus populaires et les plus utilisées pour la génération de maillage. Prenons le cas pour la 2D, dans un plan contenant N points. La triangulation comme son nom l'indique est composée de triangles dont les sommets sont les points du plan. Avec une telle description, plusieurs triangulations sont possibles :

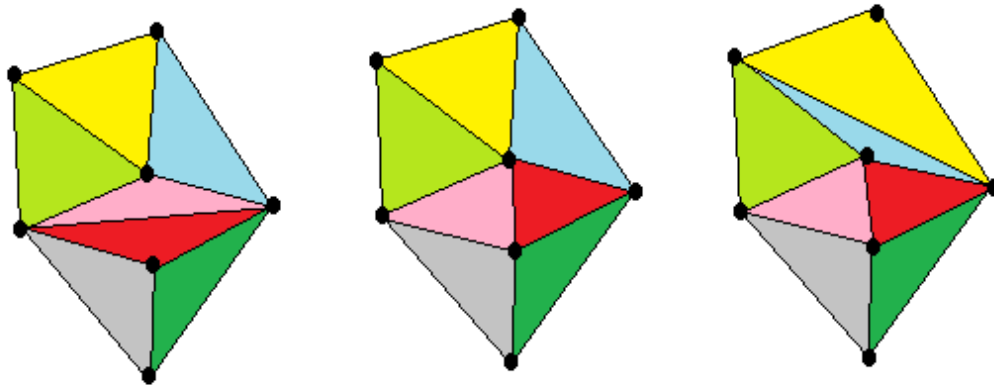


Figure 51 : exemples triangulation

3.2.3.1 Définition de la triangulation de Delaunay

Comme indiqué dans son nom, cette triangulation a été inventée par le mathématicien russe Boris Delaunay en 1934. Pour N points, un triangle de Delaunay est un triangle dont ses sommets sont trois points du plan de façon que son cercle circonscrit ne contienne aucun autre point. La combinaison des triangles ayant ces propriétés forme une triangulation de Delaunay unique. Il n'est pas possible de définir une triangulation pour des points alignés. Enfin pour des points se trouvant sur un même cercle (ex : les sommets d'un rectangle) la triangulation de Delaunay n'est pas unique car les deux diagonales satisfont les propriétés d'un triangle de Delaunay.

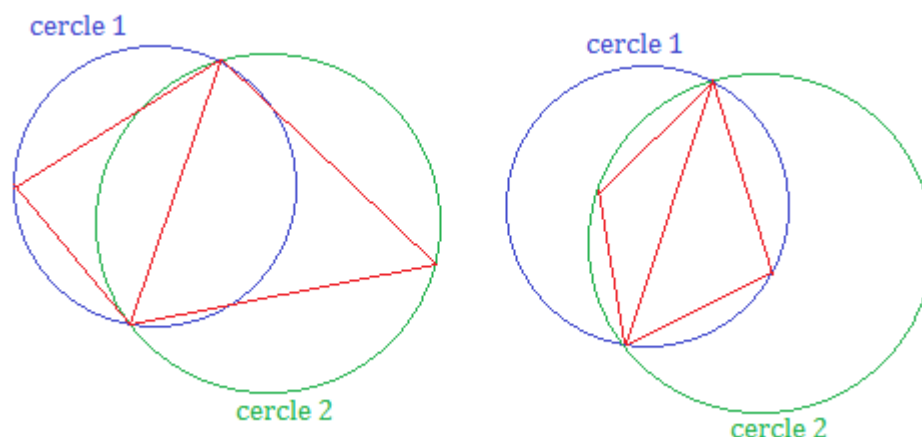


Figure 52 : triangulation avec 4 points

Voici deux triangulations avec quatre points. Celle de gauche est une triangulation de Delaunay, alors que celle de droite non. En effet les deux cercles contiennent les quatre points, cette triangulation ne remplit pas les conditions de Delaunay. Remarquons qu'avec la triangulation de Delaunay, les angles des triangles sont les plus petits possible. Cela évite ainsi comme dans le second cas les triangles trop « étirés ».

3.2.3.2 Le théorème de l'angle inscrit

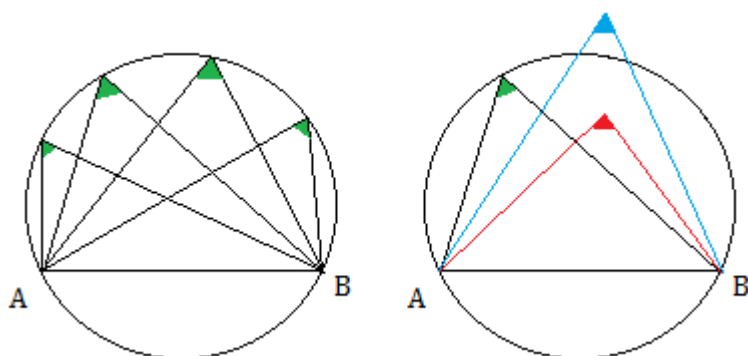


Figure 53 : angle inscrit

Pour un cercle considérons une corde AB. Le théorème de l'angle inscrit définit que quelque soit la position sur le cercle, la corde est toujours interceptée avec le même angle. Si on se place au-dessus de la corde l'angle sera aigu alors que si on est en dessous, l'angle est obtus. On appelle l'arc intercepté par la corde AB un arc capable.

Comme illustré sur la figure 53, si on considère un point à l'extérieur du cercle l'angle intercepté par la corde sera plus petit que l'angle inscrit. A contrario si on considère un point à l'intérieur du cercle, l'angle sera plus grand.

L'algorithme d'ajout de triangle utilisé est explicité en annexe 4.

3.2.3.3 Diagramme de Voronoï

Il serait difficile de parler de la triangulation de Delaunay sans aborder son lien avec le diagramme de Voronoï.

Considérons à nouveau un ensemble de N points. Il est appelé cellule de Voronoï d'un point, l'ensemble des points du plan qui appartiennent à une surface qui est la plus proche de ce point que des autres. Chaque surface définie par un polygone convexe. Le pavage d'un plan contenant N points détermine ce qu'on appelle le diagramme de Voronoï.

Soulignons que ce procédé a été inventé par un mathématicien russe G. Voronoï durant le XIXème siècle. D'autres scientifiques avaient étudié le problème précédemment dont Dirichlet (vers 1850) et plus particulièrement Descartes (1640) dans son traité sur la lumière, pour la fragmentation du cosmos. Aujourd'hui, les cellules de Voronoï sont utilisées pour l'étude de certains phénomènes naturels car elles sont assimilées à une notion d'espace vital pour certaines espèces vivantes.

Exemple avec trois points non alignés (figure 54) : les cellules sont trois zones déterminées par un angle défini par les médiatrices des côtés du triangle. L'intersection de ces droites donne le centre du cercle circonscrit.

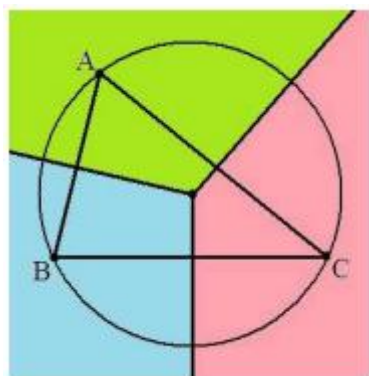


Figure 54 : Voronoï trois points non alignés

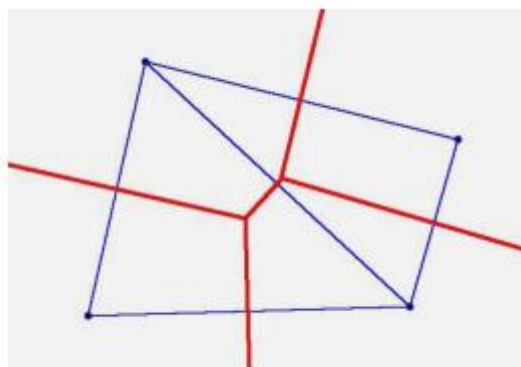


Figure 55 : Voronoï quatre points non alignés

Illustration d'une triangulation de Delaunay avec 2900 points :

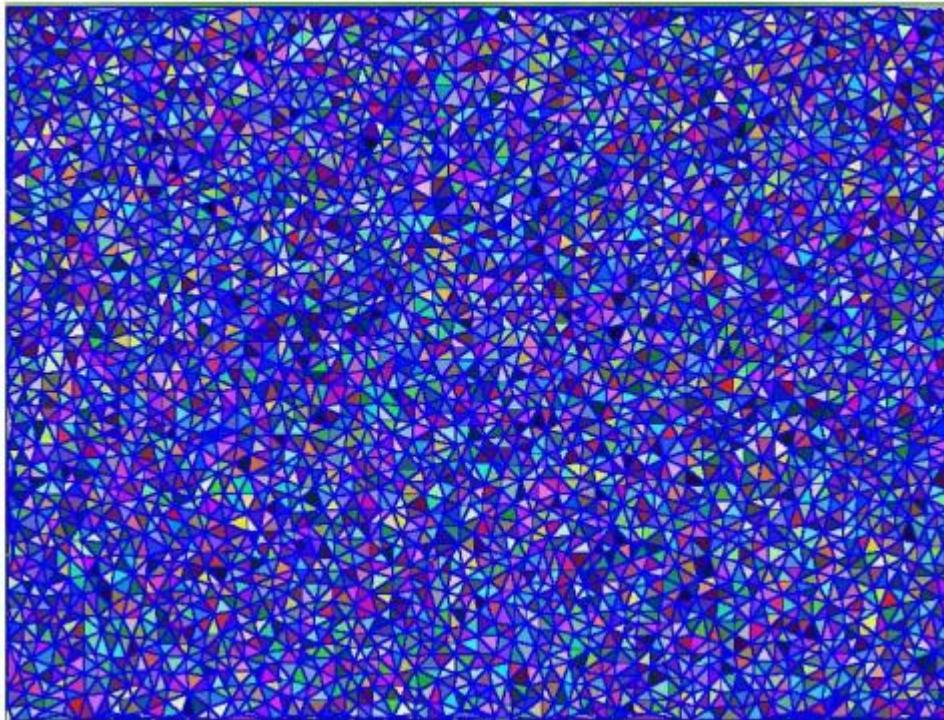


Figure 56 : triangulation de Delaunay (ai.univ-paris8.fr)

Voici le diagramme de Voronoï qui en découle :

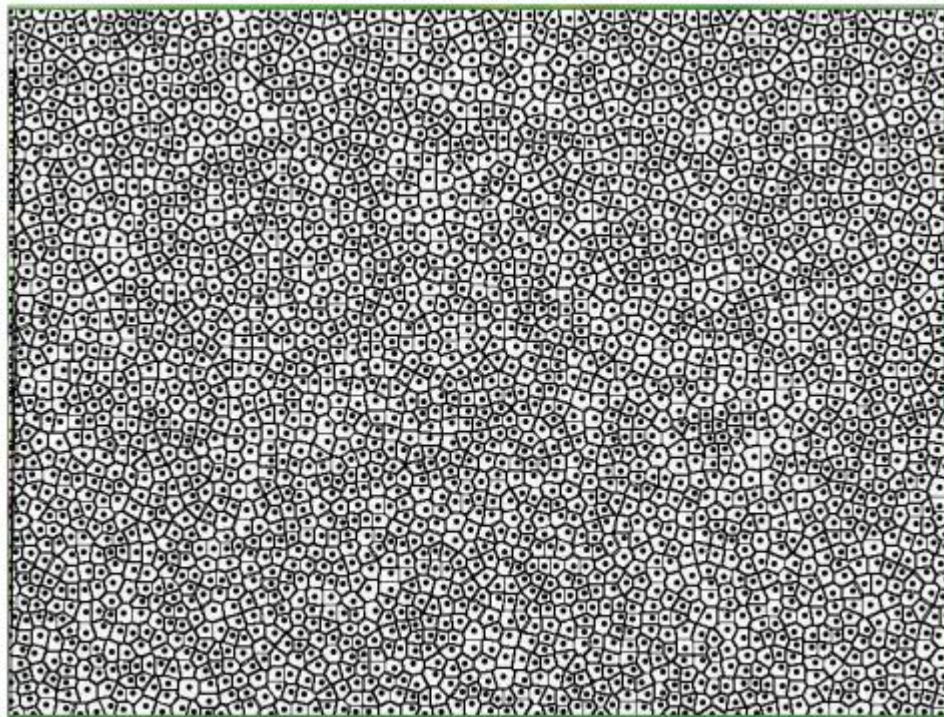


Figure 57 : diagramme de Voronoï (ai.univ-paris8.fr)

Conclusion

Tout au long de ce travail de fin d'étude, il a fallu mettre en pratique les enseignements reçus au cours des années de formation à l'ESGT. Mais pas seulement car bien qu'ayant eu des cours de programmation, il a été nécessaire de s'adapter à un nouveau langage informatique le C# (C sharp) pour réaliser cet outil informatique. Les débuts ont été quelque peu laborieux mais après quelques semaines d'apprentissage ce langage était devenu compréhensible.

D'autre part, ce projet montre la place que peut avoir le développement dans un cabinet. Il est vrai que ce n'est pas la première idée qu'on peut avoir de la profession mais de plus en plus de cabinet commence à faire des développements en interne pour faciliter et améliorer leurs méthodes de travail. Geosat, avec l'initiative de M. Lionel RAFFIN, a déjà finalisé une dizaine de plug-in pour différents logiciels.

Concernant l'outil ORTHORAMA (nom donné au plug-in crée dans SolidWorks), les objectifs de développement ont été réalisés et le résultat aujourd'hui est un gain de temps colossal par rapport à l'ancienne méthode utilisée. Pour reprendre l'élément déclencheur qui a suscité le développement de ce projet, la modélisation des bâtiments d'Airbus, le technicien travaillant sur le dossier assure que le gain de temps aurait pu être de deux à trois mois. Ceci n'étant pas négligeable car dans le monde du travail on se rend compte que le temps est une denrée précieuse pour les entreprises.

L'utilisation d'ORTHORAMA ne se fait pour l'instant que pour répondre aux demandes de modélisation 3D, ce qui ne représente pour le moment pas la majorité des clients mais dans une vision future avec la procédure de BIM (maquette 3D) cela pourra trouver une place plus importante dans les activités du cabinet.

De plus vu les résultats satisfaisants, Geosat est en contact avec les sociétés Gexcel et FARO pour parvenir à une commercialisation du plug-in sur SolidWorks. Il est tout à fait envisageable que le plug-in soit dès la fin de l'année en téléchargement sur l'App Store de FARO. En effet, les sociétés tournées vers les domaines de l'industrie ont un réel engouement pour ce produit. Ces ambitions commerciales entraînent la nécessité de faire un installateur de cette première version.

Les perspectives d'évolution :

Comme tout développement informatique, ORTHORAMA n'est qu'une version bêta et de nombreuses évolutions peuvent être effectuées. En effet, le partenariat avec des sociétés telles que Gexcel laisse imaginer la création d'un format d'image solide dédié uniquement à cet outil. Avec la possibilité de créer des images qui peuvent contenir l'information d'altitude d'une manière différente, de coder en RGB (sur trois canaux) pour pouvoir gérer les Z multiples et avoir une vraie 3D.

Une autre possibilité est d'améliorer la détection du chemin de pixel en faisant évoluer l'algorithme de Bresenham qui pour l'instant suit une ligne droite mais pourquoi pas avec des courbes.

Enfin, en ce qui concerne le MNT, sa création est vraiment basique car ce n'est qu'un réseau de triangle, ça ne représente pas concrètement la réalité, il serait donc envisageable de faire un lissage par exemple à l'aide de spline 3D.

Ce travail de fin d'étude a posé les bases d'un outil de manipulation d'images dans SolidWorks. Il a prouvé son efficacité et son utilité durant les mois d'intervention. Cependant, j'espère que la société Geosat poursuivra sur ce projet pour le faire évoluer et pourquoi pas proposer un prochain TFE pour une version 2.0.

Table des illustrations

Figure 1 : Levé scan (<i>leica-geosystems.fr</i>)	9
Figure 2 : trièdre direct	10
Figure 3 : Super Mario Bros (<i>giantbomb.com</i>)	10
Figure 4 : Courbes de niveau (<i>lezart.free.fr</i>)	11
Figure 5 : BD TOPO 3D du Havre (<i>ign.fr</i>)	11
Figure 6 : Représentation matricielle d'une image	12
Figure 7 : Image matricielle (<i>wikipédia.fr</i>)	12
Figure 8 : Relation entre l'espace image et objet	12
Figure 9 : Structure d'une image solide (voir annexe 1)	13
Figure 10 : Exemple de tronc (<i>wikipédia.fr</i>)	13
Figure 12 : Vue en perspective (<i>wiki.povray.org</i>)	14
Figure 11 : Vue orthographique (<i>wiki.povray.org</i>)	14
Figure 13 : Images de la pyramide (<i>cours Polimi</i>)	14
Figure 14 : Pyramide d'image (<i>cours Polimi</i>)	14
Figure 15 : Paramétrage d'export (<i>Gexcel</i>)	15
Figure 16 : Exemple utilisation plug-in AutoCAD (<i>Gexcel</i>)	15
Figure 17 : Exemple carte (<i>loisir.ign.fr</i>)	16
Figure 18 : Exemple carte (<i>loisir.ign.fr</i>)	16
Figure 19 : Système TLS (<i>commentcamarche.net</i>)	16
Figure 20 : Carte de France (<i>cartefrance.fr</i>)	17
Figure 21 : logiciel PointR3, planarity map	17
Figure 22 : JRC 3D Reconstructot (<i>gexcel.it</i>)	19
Figure 23 : Rééchantillonnage (<i>zabaque.upac.ca/infographie</i>)	22
Figure 24 : codage RLE (<i>wikipédia.fr</i>)	22
Figure 25 : Origine de l'esquisse	23
Figure 26 : Vue normale au plan	23
Figure 27 : Arbre de SolidWorks	24
Figure 28 : Exemple XML	26
Figure 29 : XML propre aux ortho-images	27
Figure 30 : cité administrative de Bordeaux (<i>pss-archi.eu</i>)	28
Figure 31 : résultat obtenu avec SolidWorks	28
Figure 32 : rotation	28
Figure 33 : translation	28
Figure 34 : pixels voisins	28
Figure 35 : exemples tracés	28
Figure 36 : procédure de sélection des pixels	28
Figure 37 : Division en sections	28
Figure 38 : Sélection du pixel	28
Figure 39 : chemin de pixel	28
Figure 40 : projeté orthogonal	28
Figure 41 : chemin projeté sur la ligne	28
Figure 42 : image solide	28
Figure 43 : droite en 3D (vue de côté)	28
Figure 44 : droite en 2D	28

Figure 45 : volume en 3D de l'escalier	28
Figure 46 : Exemple MNT (<i>caillie13.btstopo.free.fr</i>)	28
Figure 47 : MNT, MNS et MNE (<i>ensg.eu</i>)	28
Figure 48 : MNT (vue perspective)	28
Figure 49 : MNT (vue de face).....	28
Figure 50 : points du MNT.....	28
Figure 51 : exemples triangulation.....	28
Figure 52 : triangulation avec 4 points	28
Figure 53 : angle inscrit.....	28
Figure 54 : Voronoï trois points non alignés	28
Figure 55 : Voronoï quatre points non alignés	28
Figure 56 : triangulation de Delaunay	28
Figure 57 : diagramme de Voronoï.....	28

Bibliographie et référence web

BIAGI L. (sous la coordination de)(2014). *Support de cours – Géodésie*, Polytechnique de Milan, seconde année de Master en Sciences Ingénierie Géomatique.

CALI J. et LABERGERIE E. et NICOLAS J. et SIMONETTO E. (sous la coordination de)(2013). *Support de Cours – Option Relevé 3D*, ESGT ES3.

HAPPE D. (2011). *La réalité augmentée au service de la mise en valeur du patrimoine culturel*. XYZ – Revue de l'Association Française de Topographie, 128 : 50-54.

HILAIRE N. (2012). *Apprenez à développer en C#, La programmation en C# .Net expliquée aux débutants*. Simple It, Livre Du Zero, 490p.

LANDES, T. et GRUSSENMEYER, P. (2011). *Les principes fondamentaux de la lasergrammétrie terrestre: systèmes et caractéristiques*. XYZ - Revue de l'Association Française de Topographie, 128 : 37-49.

LANDES, T. et GRUSSENMEYER, P. et BOULAASSAL, H. (2011). *Les principes fondamentaux de la lasergrammétrie terrestre: acquisition, traitement des données et applications*. XYZ - Revue de l'Association Française de Topographie, 129 : 25-38.

MALPASS L. (2011). *SolidWorks API Programming and Automation*. AngelSix, 268p.

MALPASS L. (2011). *SolidWorks API Advanced Product development*. AngelSix, 246p.

NARVARTH (2012). *Apprenez à programmer en C# sur .NET*. OpenClassrooms, 131p.

SCHUHMACHER S. et BÖHM J. (). Georeferencing of terrestrial lasercanner data for applications in architectural modeling. University of Stuttgart, Institute for Photogrammetry, Germany, 7p.

SONNA G. (sous la coordination de)(2014). *Support de cours – Télédétection et Analyse d'image*, Polytechnique de Milan, seconde année de Master en Sciences Ingénierie Géomatique.

UEBERSCHLAG A., GILIERON P.Y., ZUFFEREY J. (2012). *Des millions de points pour une modélisation du réel*. Flash informatique n°6.

API Support SolidWorks : www.solidworks.com>Support (consulté tout au long du projet)

Forum API SolidWorks : forum.solidworks.com/community/solidworks/api, (consulté tout au long du projet)

Gexcel : www.gexcel.it/en, (consulté en mars)

Faro Scene : www.faro.com/fr-fr/produits/logiciels-faro/scene, (consulté en mars)

Infographie : raphaello.univ-fcomte.fr/lg/Algorithmt/Algorithme.htm, (consulté en avril)

PointCab : www.pointcab-software.com/en/, (consulté en mars)

Site du zéro : fr.openclassrooms.com, (consulté tout au long du projet)

Table des annexes

Annexe 1 : A new concept : the solid image	p.49
Annexe 2 : Gexcel JRC 3D Reconstructor	p.56
Annexe 3 : Exemple d'utilisation des ortho-images dans SolidWorks	p.60
Annexe 4 : Algorithme de Bresenham	p.62
Annexe 5 : Algorithme de la triangulation de Delaunay	p.64
Annexe 6 : Poster de présentation	p.65

Annexe 1 : A new concept : the solid image

Annexe 2 : Gexcel JRC 3D Reconstructor



Source : http://www.gexcel.homeip.net/Reconstructor/R_News/R_News_Jun2014/JRC3DReconstructor2.9.2_WhatIsNew.pdf

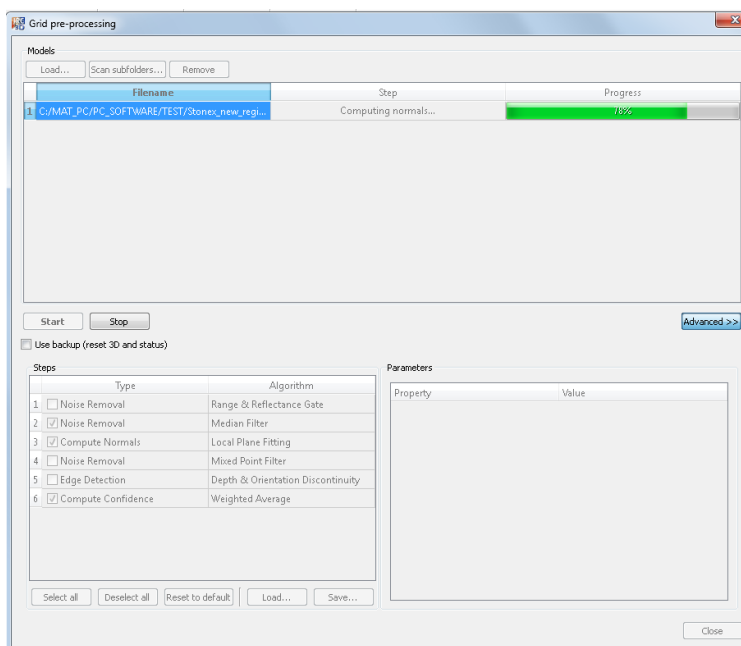
JRC 3D Reconstructor® 2.9.2 What is new?

The new release of Gexcel's **JRC 3D Reconstructor® 2.9.2** offers important improvements for faster and lighter data processing, with revolutionary point merging filters and alignment tools, keeping in step with the latest technologies in 3D laser scanning and photogrammetry.

- **Japanese software translation**

JRC 3D Reconstructor is now available also in **Japanese** along with English, Chinese and Italian.

- **Faster and lighter version**



1. **Twice faster pre-processing** step like **data filtering** (to rapidly remove noise, calculate points quality and inclination).

2. JRC 3D Reconstructor **projects** now **lighter** for a more efficient data storage. Function fully retro-compatibility guaranteed for old projects.



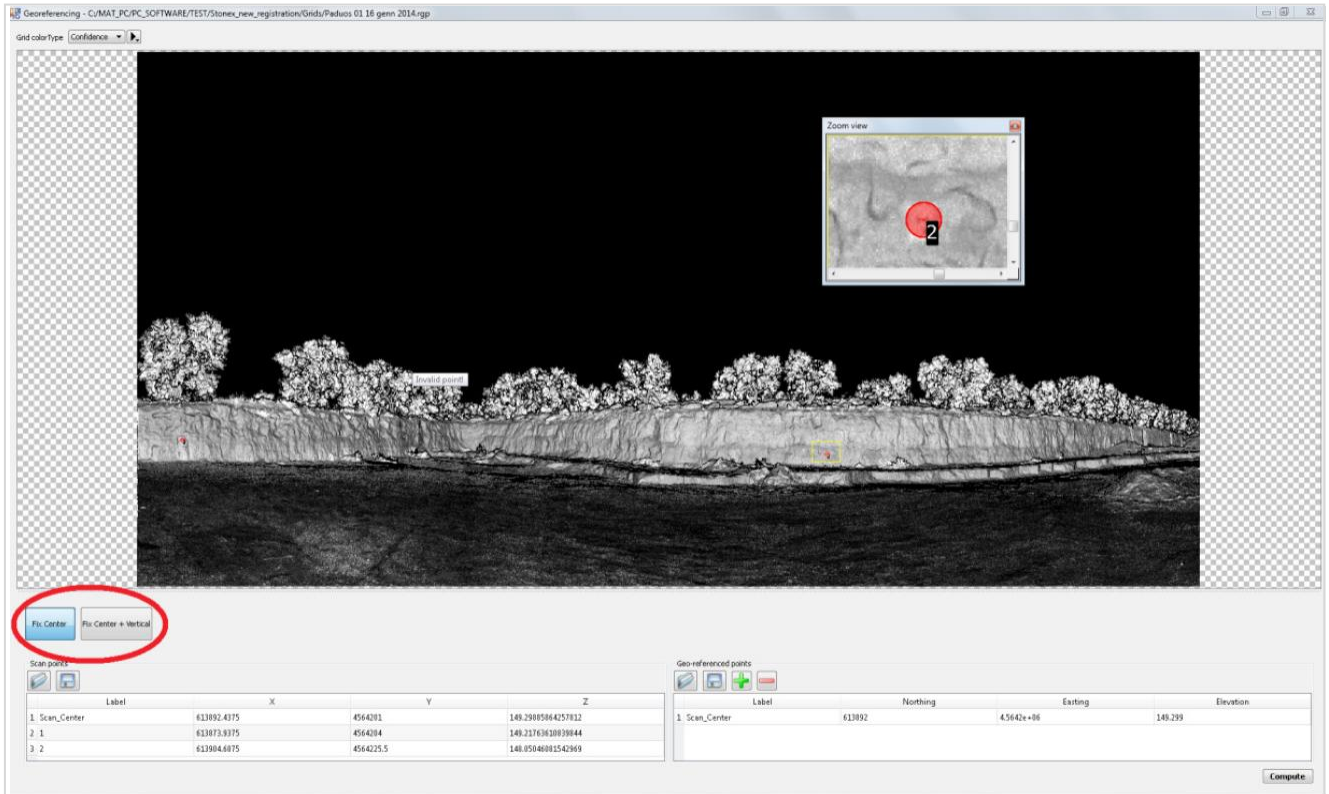
3. **Saving space tool** to free up your hard disk by compressing your old JRC 3D Reconstructor projects.

A new script function can rapidly compress your projects in the new 2.9.2 format.

4. **Merging tool** for JRC 3D Reconstructor **projects** to help parallel data processing. A project can be subdivided between more users and merged in a second time.

- **Improvements in Registration**

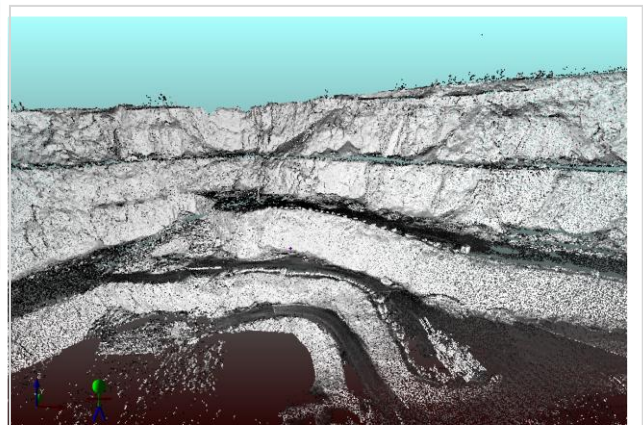
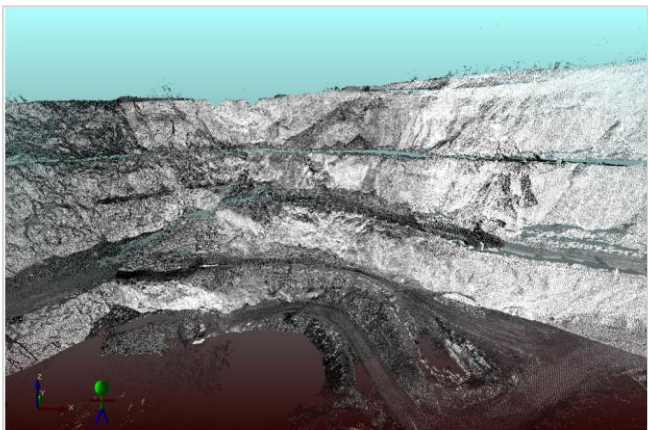
Faster geo-referencing tool by using GPS or scan position and 2 or 3 external points.



Faster scans pre-registration in two clicks. If the scan position is known, two points are required to pre-align the scans, only one point if scan position and verticality are fixed.

- **Revolutionary merging filter**

The new *Level 3D density of point clouds* filtering allows you to merge large point clouds by choosing the best points in the overlapping areas

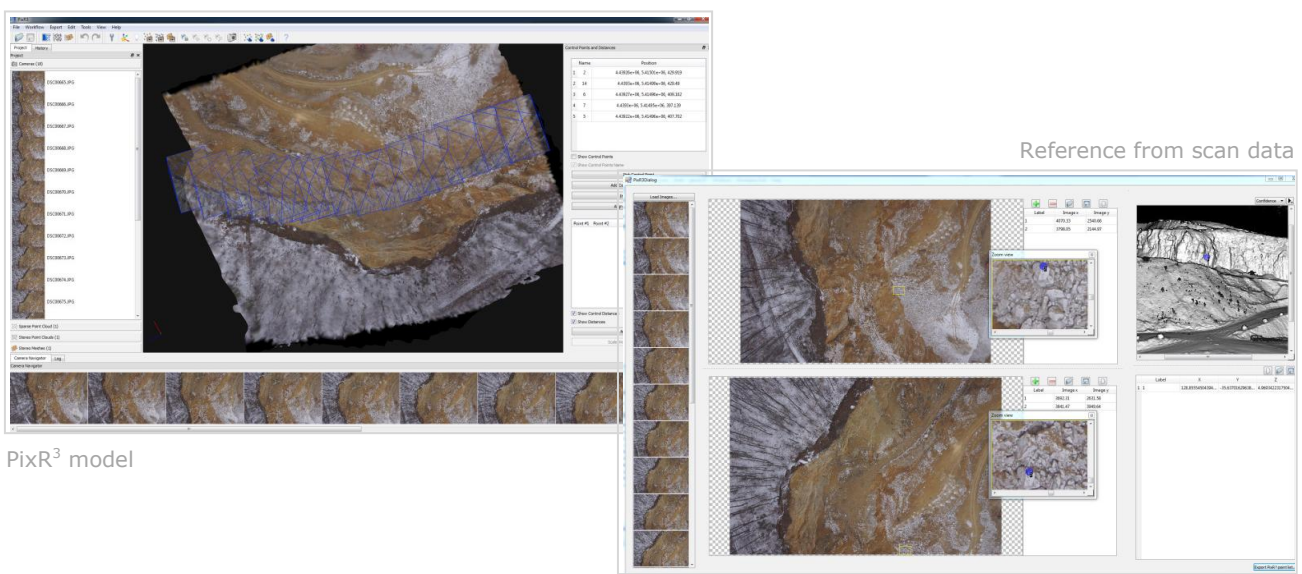


- **Scans coloring improved**

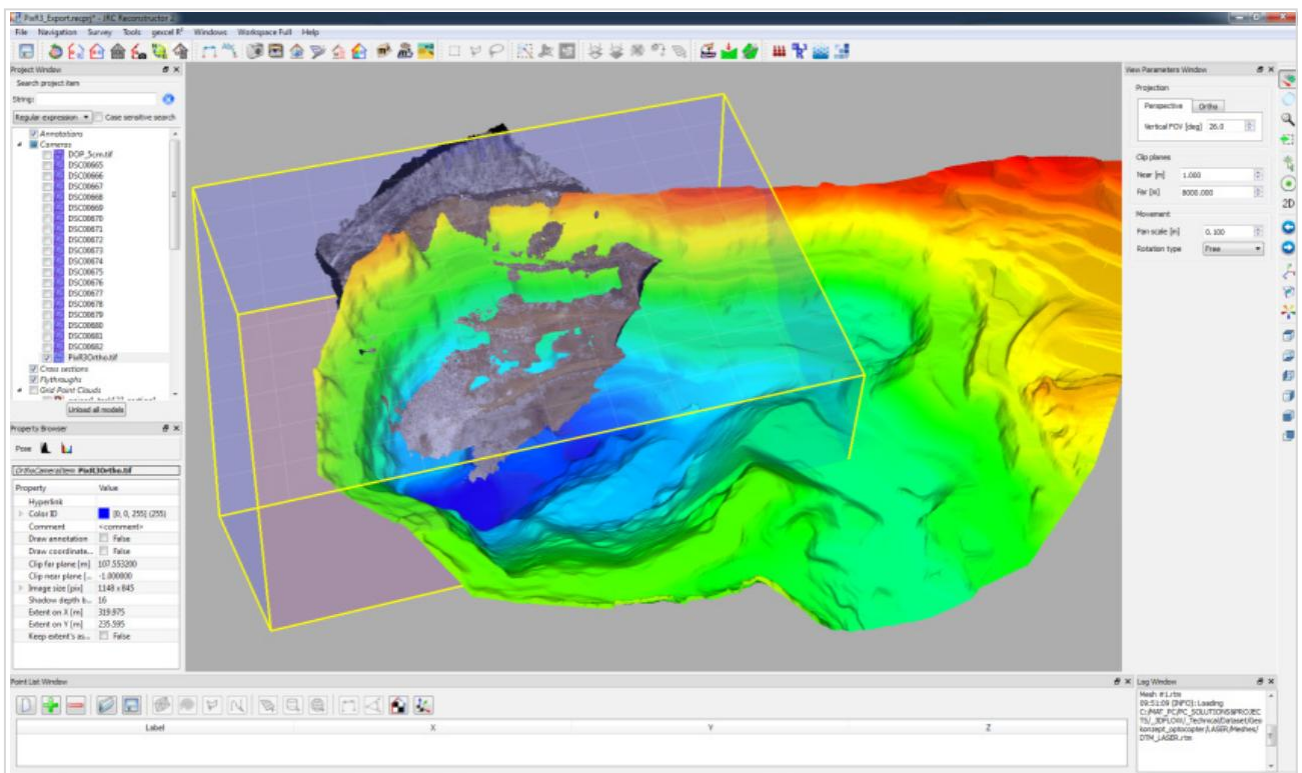
Now you can use the **external cameras** to color your scans without parallax effects. The scans are colored by considering the points of view differences between scans and images.

- **JRC 3D Reconstructor and PixR³**

Fully integration to merge **photogrammetric models** from **UAV** or imagery with lidar data sets. You can scale and reference your photogrammetric models according to lidar data and merge them with points, 3D models and oriented images in JRC 3D Reconstructor.



Imagery mode connected to a lidar based one



- **Lidar sensors** supported
 - Import of **Z+F projects** by supporting the new high resolution images for 5010c
 - Dedicated importing of **Stonex X300** format, with the new automatic coloring tools
 - **ReLinkAPP** for Scene 5.1, 5.2 and 5.3 (beta version)*
 - Improved the **Optech ILRIS Parser and Controller** with new online help
 - Improved the support for **LAS** and **E57** data
 - Improved the point clouds in **ASCII format** with scaling capabilities
 - Updated the mesh import functions with the new **WRML** library

*Two versions available:

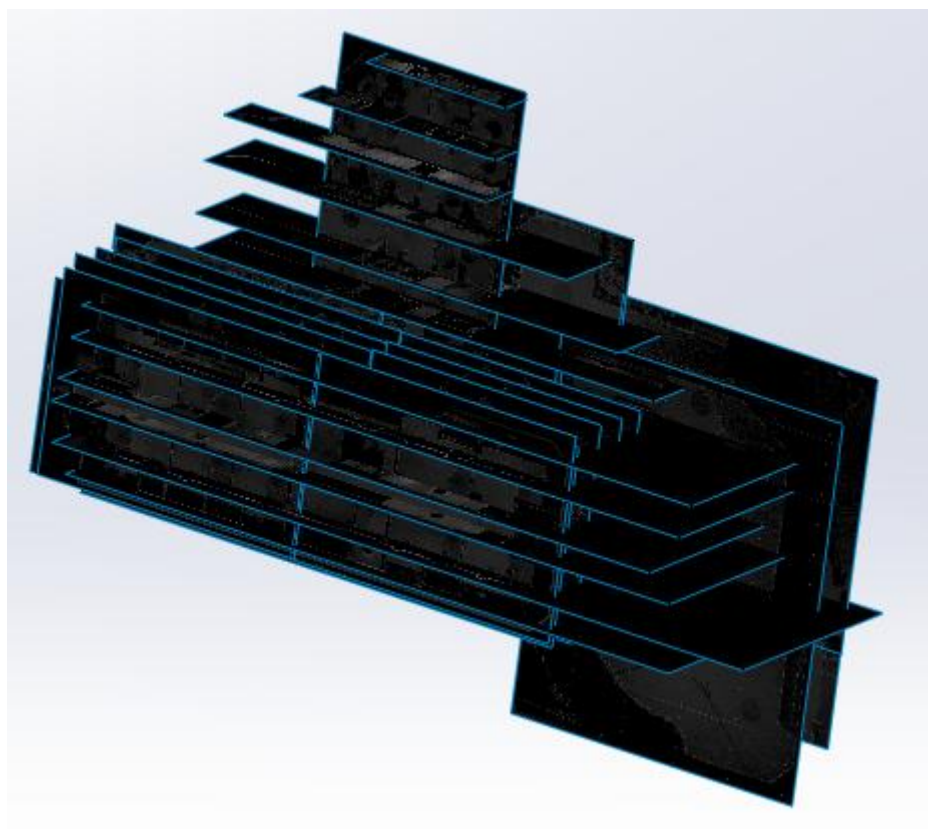
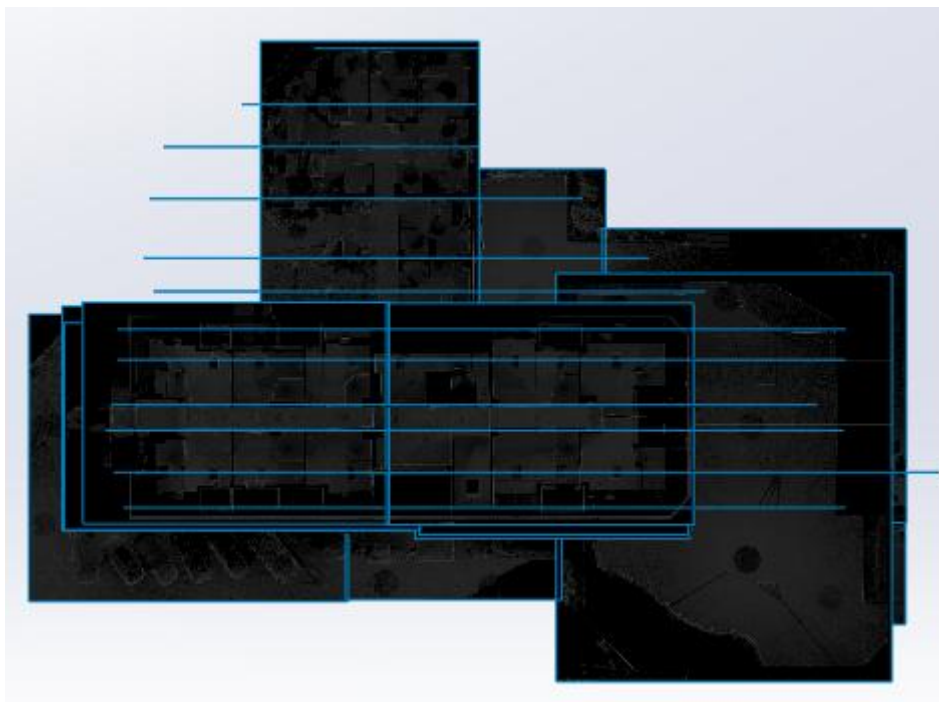
- for JRC 3D Reconstructor 2.9.1 users or older
- for JRC 3D Reconstructor 2.9.2 users with the new compressed file versions

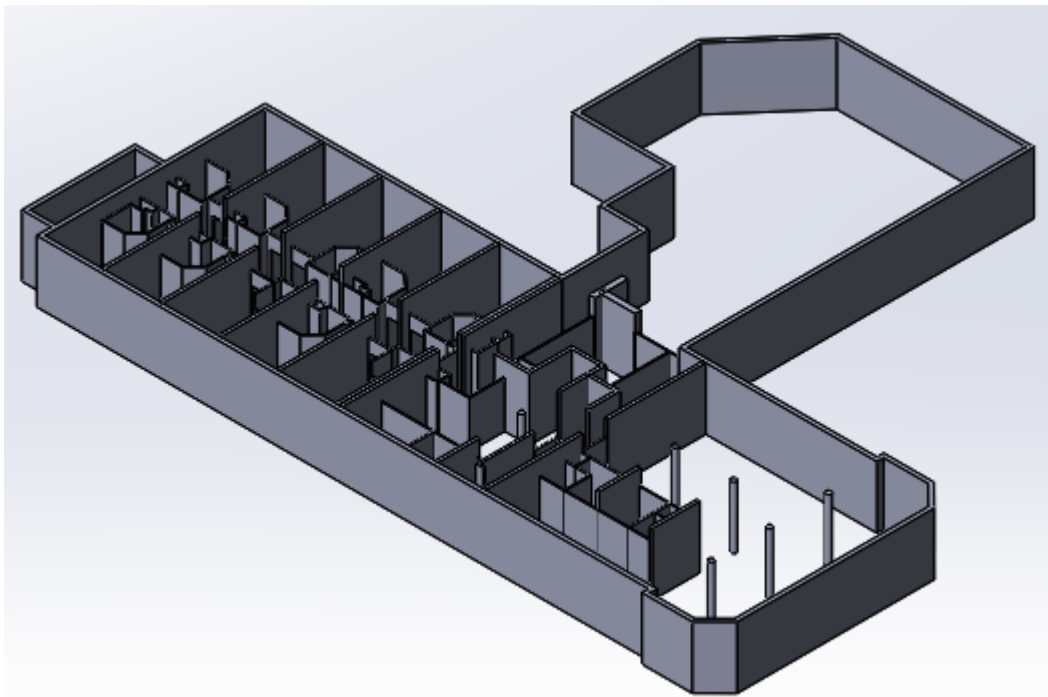
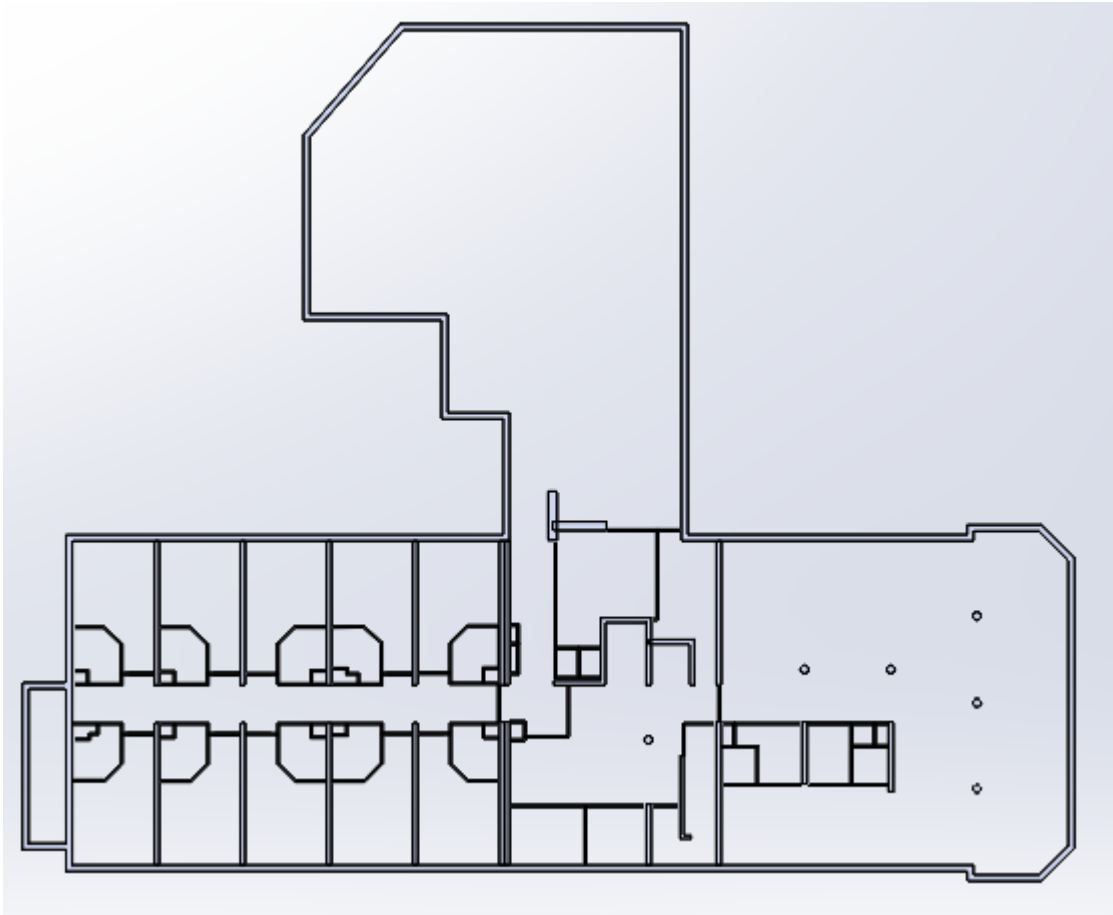
Completely free you can immediately view all your data in the JRC 3D Reconstructor

These many improvements make the Gexcel software family unprecedented in advanced **functionality, flexibility** and **ease of use**. Inquire about the regular **webinars** and activities Gexcel organizes for clients and users worldwide: sales@gexcel.it.

Annexe 3 : Exemple d'utilisation des ortho-images dans SolidWorks

Voici un autre exemple d'utilisation des ortho-images avec SolidWorks. Un bâtiment doit être modélisé en 3D. Pour cela des coupes et des plans sont réalisés grâce aux logiciels de traitement de nuage de points. Puis à l'aide du plug-in développé les ortho-images sont insérées dans le logiciel SolidWorks automatiquement géoréférencées. Le résultat est donné par les illustrations suivantes : une sorte de « millefeuille » d'ortho-images avec les coupes et les plans qui se croisent perpendiculairement.





Annexe 4 : Algorithme de Bresenham

Voici l'algorithme utilisé pour la détection des chemins de pixels :

- 1- Avec l'équation cartésienne de la ligne
- 2- Avec l'algorithme de Bresenham

Algorithme 1 :

Les variables sont les suivantes :

- X_i, Y_i : les coordonnées du point initial du segment
- X_f, Y_f : les coordonnées du point final du segment

L'équation cartésienne est : $Y = AX+B$ avec $A = \frac{Y_f - Y_i}{X_f - X_i}$ et $B = Y_i - AX_i$

```
void ligne(int xi,int yi,int xf,int yf) {
    int x,y ;
    double a,b ;
    a =(double) (yf-yi)/(xf-xi) ;
    b = yi - a * xi ;
    for ( x = xi ; x <= xf ; x++ ) {
        y =(int) (a * x + b) ;
        allume_pixel(x,y) ; }
}
```

→ On allume tous les pixel qui ont pour coordonnées (x,y) .

Algorithme 2 : Bresenham

Les variables de coordonnées sont les mêmes que dans l'algorithme précédent mais on ajoute une nouvelle CUMUL pour conserver la valeur de la différence entre l'ordonnée entière en l'ordonnée réelle.

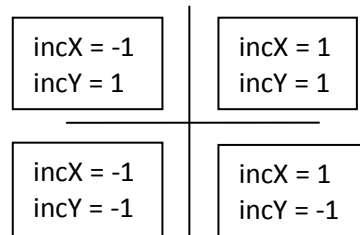
Les hypothèses sont les suivantes : $X_f > X_i, Y_f \geq Y_i, X_f - X_i > Y_f - Y_i$

```
void ligne(int xi,int yi,int xf,int yf) {
    int dx,dy,cumul,x,y ;
    x = xi ;
    y = yi ;
    dx = xf - xi ;
    dy = yf - yi ;
    allume_pixel(x,y) ;
    cumul = dx / 2 ;
    for ( x = xi+1 ; x <= xf ; x++ ) {
        cumul += dy ;
        if ( cumul >= dx ) {
            cumul -= dx ;
            y += 1 ; }
        allume_pixel(x,y) ; }
}
```

- ➔ On allume tous les pixels qui ont une abscisse entière X comprise entre Xi et Xf.
- ➔ Le dernier pixel sélectionné est le pixel final car la variable CUMUL a pour valeur à la fin de l'exécution : $CUMUL = \frac{dx}{2} + dy \times dx$

Algorithme 3 : Bresenham pour tout segment

Deux dernières variables sont ajoutées pour gérer l'incrémentation en X et Y suivant le coefficient directeur du segment :



```
void ligne(int xi,int yi,int xf,int yf) {
    int dx,dy,i,incx,incy,cumul,x,y ;
    x = xi ;
    y = yi ;
    dx = xf - xi ;
    dy = yf - yi ;
    incx = ( dx > 0 ) ? 1 : -1 ;
    incy = ( dy > 0 ) ? 1 : -1 ;
    dx = abs(dx) ;
    dy = abs(dy) ;
    allume_pixel(x,y) ;
    if ( dx > dy ) {
        cumul = dx / 2 ;
        for ( i = 1 ; i <= dx ; i++ ) {
            x += incx ;
            cumul += dy ;
            if ( cumul >= dx ) {
                cumul -= dx ;
                y += incy ;
            }
            allume_pixel(x,y) ; } }
    else {
        cumul = dy / 2 ;
        for ( i = 1 ; i <= dy ; i++ ) {
            y += incy ;
            cumul += dx ;
            if ( cumul >= dy ) {
                cumul -= dy ;
                x += incx ;
            }
            allume_pixel(x,y) ; } }
}
```

Annexe 5 : Algorithme de la triangulation de Delaunay

Afin de ne pas mettre des pages entières de code dans un langage spécifique, il m'a paru plus approprié de décrire l'algorithme de la triangulation de Delaunay en pseudo-code pour que chacun puisse l'adapter à son utilisation.

Voici le pseudo code permettant d'ajouter un point à la triangulation de Delaunay :

Donnée d'entrée : liste des sommets

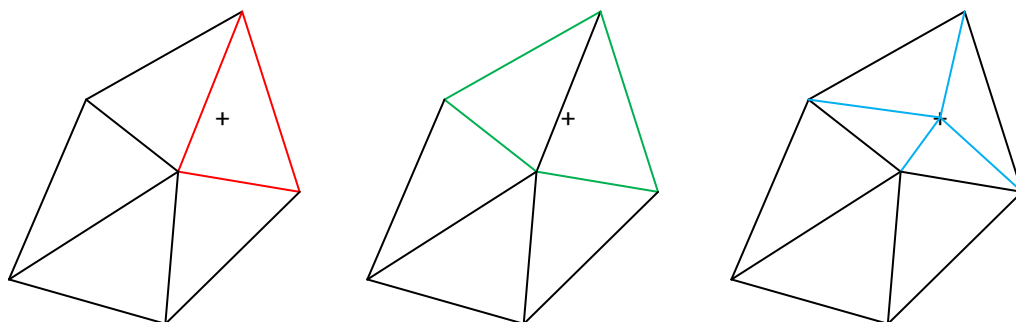
Donnée de sortie : liste de triangles

Triangulation

```

{
  Initialiser de la liste de triangles
  Déterminer le « super-triangle »
  Ajouter les sommets du « super-triangle » à la fin de la liste des sommets
  Ajouter le « super-triangle » à la liste des triangles
  Pour chaque point de la liste des sommets
  {
    Initialiser la mémoire
    Pour chaque triangle présent dans la liste de triangle
    {
      Calculer le centre et le rayon du cercle circonscrit
      Si le point se trouve dans le cercle circonscrit
      {
        Alors ajouter les trois bords du triangle à la mémoire
        Supprimer le triangle de la liste de triangle
      }
    }
    Supprimer tous les côtés des triangles qui sont en double dans la mémoire //il ne
    reste à la fin que les bords du polygone
    Ajouter à la liste des triangles tous les triangles formés entre le premier point et
    les bords du polygone englobant
  }
  Supprimer tous les triangles de la liste de triangles qui utilisent les sommets du « super-
  triangle »
  Supprimer les sommets du « super-triangle » de la liste des sommets
}
  
```

Le super-triangle en rouge, le polygone en vert et la nouvelle triangulation en bleue.



Réalisation d'un plug-in de manipulation d'images 3D au sein de l'environnement SolidWorks

Mémoire de Travail de Fin d'Etudes
Blandine GANIVET

Résumé :

SolidWorks, logiciel dédié principalement à la mécanique permet de réaliser des modélisations en trois dimensions. Or il n'est pas adapté au traitement des levés scanners et n'est pas apte à gérer les nuages de points. Une méthode a donc été imaginée au sein du cabinet mais sa durée d'exécution était trop importante. Pour pallier à cet obstacle, la solution de développer un plug-in pour le logiciel a vu le jour. A partir des levés effectués sur le terrain des ortho-images sont créées par des logiciels de traitement tel que SCENE, PointCab ou Gexcel. L'objectif du plug-in est de pouvoir intégrer ces ortho-images dans SolidWorks directement géoréférencée afin de pouvoir modéliser en trois dimensions.

Dans un second temps, s'appuyant sur une ortho-image contenant l'information d'altitude dans ces pixels, il sera possible d'obtenir une modélisation 3D en seulement quelques clics.

Mots clés : Plug-in, SolidWorks, Ortho-image, Modélisation 3D, Image solide

Abstract :

SolidWorks, software usually dedicated to mechanic, allows to realize three-dimensional modeling. But it is not suitable for scanners measure treatment and it is not able to manage point clouds. Thus a method was conceived by the company but its execution time was too long. To overcome this obstacle, the solution to develop a plug-in software emerged. Based on surveys realized in the field, ortho-images are created by processing software such as SCENE, PointCab or Gexcel. The plug-in aim is to integrate these ortho-images in SolidWorks directly georeferenced in order to model in three dimensions.

In a second step, based on ortho-image containing altitude information in its pixels, it is possible to obtain a 3D modeling in few clicks.

Key words : Plug-in, SolidWorks, Ortho-image, 3D Modeling, Solid Image