



**HAL**  
open science

# Analyse de visibilité et géolocalisation en milieu urbain avec un modèle numérique 3D

Wassim Suleiman

► **To cite this version:**

Wassim Suleiman. Analyse de visibilité et géolocalisation en milieu urbain avec un modèle numérique 3D. Géographie. Université Jean Monnet - Saint-Etienne, 2013. Français. NNT : 2013STET2187 . tel-01060183

**HAL Id: tel-01060183**

**<https://theses.hal.science/tel-01060183v1>**

Submitted on 3 Sep 2014

**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.

UNIVERSITE Jean Monnet Saint-Etienne

**THESE**

Pour obtenir le grade de  
*DOCTEUR DE L'UNIVERSITE Jean Monnet*

*Spécialité* : Gestion de l'information géographique

Présentée et soutenue publiquement  
Par

Wassim SULEIMAN

Le 25/09/2013

**Analyse de visibilité et géolocalisation en milieu urbain  
avec un modèle numérique 3D**

Directeurs de thèse : Thierry JOLIVEAU et Eric FAVIER

**Jury**

Professeur Bin Jiang, Department of Technology and Built Environment, University of Gävle, Suède.

Professeur Guillaume Moreau, Ecole Centrale de Nantes, France. Rapporteur

Professeur Serge Miguet, Université Lumière Lyon 2, France. Rapporteur

Professeur Alain Tremeau, Université Jean Monnet, France.

Professeur Thierry Joliveau, Université Jean Monnet, France. Directeur de thèse



## REMERCIEMENTS

Je remercie très chaleureusement mon directeur de thèse, Thierry JOLIVEAU, pour son aide très précieuse. Son regard à la fois critique et bienveillant sur mes textes, sa culture, les discussions que nous avons pu avoir lors de nos nombreuses rencontres.

Je remercie également mon co-directeur de thèse Eric FAVIER, qui est à l'origine de tout ceci. Son soutien et ses conseils ont été sources d'enrichissement de ma réflexion. Son décès a été un choc pour nous tous et une grande perte pour la communauté scientifique à l'ENISE. Je remercie M. Alain TREMEAU d'avoir accepté de relire le manuscrit.

Je tiens à remercier Guillaume Moreau, Serge Miguet pour avoir rapporté sur ma thèse et Bin Jiang et Alain Tremeau d'avoir accepté d'évaluer mon travail. Je n'oublie pas d'exprimer ma reconnaissance à mes amis Pierre Olivier Mazagol, Laurent Lafuente, Clément Dodane et Florent Fayeau pour leur aide précieuse.

*Je dédie ce mémoire à ma famille, mon épouse Ola et mon enfant Hussin, qui m'ont accompagné tout au long de ces années de travail intense.*



## Table des matières

Introduction générale.....	9
Chapitre 1 Problématique, concepts et méthodes du domaine.....	11
1.1 Problématique d'ensemble .....	11
1.2 Reconnaissance des bâtiments dans une image et géolocalisation de la prise de vue.....	18
1.2.1 Reconnaissance des bâtiments par l'appariement entre l'image de synthèse et l'image réelle.....	19
1.2.2 Reconnaissance des bâtiments à partir de leurs textures.....	20
1.2.3 La géolocalisation à partir des images .....	22
1.3 L'isovist.....	23
1.3.1 Calcul de l'isovist en 2 dimensions.....	24
1.3.2 Les programmes de calcul du champ de la visibilité en dimension 2 .....	27
1.3.3 Les méthodologies de calcul du champ de la visibilité en dimension 3 .....	27
1.3.4 La syntaxe spatiale (space syntax) .....	30
1.4 Conclusion.....	31
Chapitre 2 Reconnaissance de bâtiments et localisation de photographies au moyen d'un descripteur de texture .....	33
2.1 L'enrichissement de la base des données .....	33
2.2 La reconnaissance des façades des bâtiments dans une photographie quelconque... ..	36
2.3 La géolocalisation d'une prise de vue .....	41
2.4 Conclusion et pistes pour des travaux futur.....	45
Chapitre 3 Analyse de la visibilité urbaine par nuages des points 3D .....	49
3.1 Environnement 3D vectoriel.....	49
3.2 Maillage des points et calcul d'intervisibilité.....	51
3.3 Calcul du champ de la visibilité .....	55
3.4 L'effet d'un bâtiment sur l'intervisibilité .....	56
3.5 Evaluation du résultat .....	58
Simplification du MNT .....	59
Simplification des polygones de bâtiment .....	59
3.6 Conclusion.....	60
Chapitre 4 Une nouvelle méthode pour calculer le champ de vision avec des données vectorielles de dimension 2 et 3 .....	61
4.1 Nouvel algorithme pour calculer le champ de visibilité en dimension 2 .....	61
4.2 Un nouvel algorithme pour calculer le champ de visibilité en dimension 3 .....	71
4.3 L'application dans un environnement 3D .....	73
4.3.1 La combinaison avec les photographies.....	75
4.3.2 La syntaxe spatiale .....	76

4.4 Conclusion .....	78
Conclusion générale .....	79
Références .....	81
Références du chapitre 1 .....	81
Références du chapitre 2 .....	85
Références du chapitre 3 .....	86
Références du chapitre 4 .....	87
Publications scientifiques .....	88







# Introduction générale

Avec le développement des technologies numériques, on trouve de plus en plus d'appareils ayant une fonction de prise de vue et de GPS. Ces appareils offrent la possibilité de localiser directement des images dans un système de cartographie comme Picasa, Panoramio ou Flickr. Ces services permettent aux utilisateurs de gérer leurs photographies en fonction de l'endroit visité et de les partager avec d'autres usagers en les rendant publiques (Figure 1).

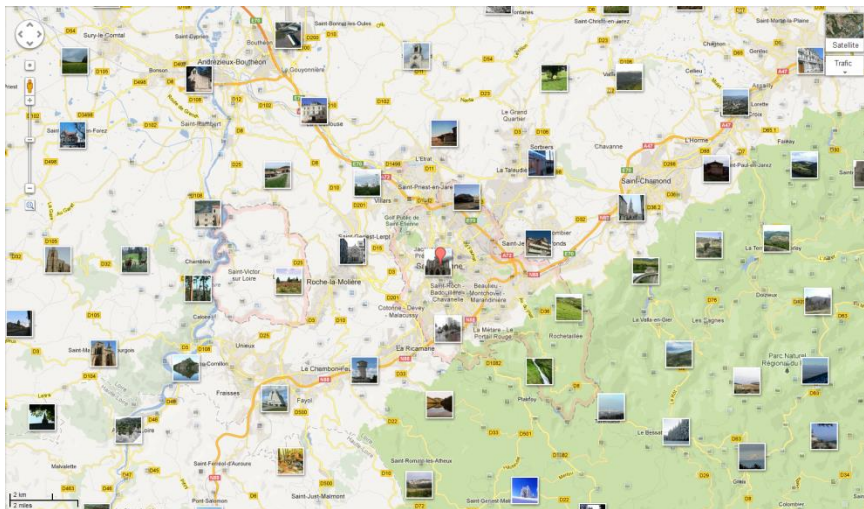


Figure 1. les photographies de Panoramio visibles dans Google Map

Cette multiplication d'images photographiques localisées, qui sont autant de vues spatiales ou géographiques, ouvrent de nouvelles applications mettant en relation l'espace vu et la photographie. Il devient par exemple possible d'utiliser la géométrie du contenu de l'image pour essayer de déterminer le point de prise de vue, si celui-ci n'est pas connu. C'est un domaine de recherche récent et en cours de développement. On peut aussi inverser la problématique et tenter de décrire l'espace géographique visible depuis le point de vue où est prise la photographie. Dans la littérature, cette question est celle du calcul de l'isovist ou de l'analyse de la visibilité. C'est une question de recherche importante qui peut trouver des applications dans de nombreux domaines. La question de la délimitation automatique de l'espace visible d'un observateur intéresse par exemple le domaine de la sécurité pour le placement des caméras de surveillance dans une rue, une banque ou un musée. Cela peut être utile dans les télécommunications pour un placement optimisé de bornes wifi afin d'obtenir une bonne couverture avec un nombre minimum de bornes. Dans le domaine de la publicité

urbaine un outil d'analyse de visibilité peut aider à localiser les endroits les plus favorables à l'installation de panneaux ou d'enseignes lumineuses. L'analyse de la visibilité est centrale aussi dans le domaine de la gestion des paysages et de l'aménagement urbain, qu'il s'agisse de l'analyse des impacts paysagers ou de la mise en œuvre d'accès piétonniers ... On sait enfin que l'accès à une vue est un facteur important de choix des localisations résidentielles, qu'il s'agisse d'une vue sur l'eau, comme observé par Luttik (2000), sur la forêt (Gueymard 2006) ou sur le paysage en général (Jim et Chen 2006). Robert (2011) a ainsi bien mis en évidence l'importance de la vue sur mer dans l'urbanisation du littoral de la Côte d'Azur. L'effet sur les prix immobiliers de la nature des vues a aussi été démontré par plusieurs études Miller (2001), Cavailhès, Brossard, Hilal, Joly, Tourneux, Tritz et Wavresky (2007). Qu'il s'agisse de la question de la géolocalisation automatique d'une prise de vue numérique à partir d'images ou du calcul d'un isovist dans un environnement numérique, de nombreuses questions informatiques ne sont pas résolues. S'il existe par exemple plusieurs méthodes de calcul d'isovist en 2D, son calcul dans un environnement numérique 3D est une question plus complexe.

Cette recherche vise à donner des réponses originales et opérationnelles à ces deux questions dans le contexte des Sciences de l'Information Géographique. Il s'agit d'adapter des solutions informatiques existantes à des données plus proches de situations réelles observables que celles par rapport auxquelles ces solutions avaient été élaborées au premier chef.. Ce mémoire comporte quatre chapitres. Le premier est une introduction au problème sur la base des travaux réalisés dans le domaine. Il présente la problématique et pointe les avantages et les limites des travaux existants Le deuxième chapitre présente la méthode de reconnaissance des bâtiments et la géolocalisation des prises de vue. Les tests conduits s'appuient sur un échantillon de prises de vues de la ville de Saint Etienne. Le troisième chapitre traite de la méthode de calcul de la visibilité qui utilise un nuage de points 3D. Le quatrième chapitre expose la nouvelle méthode développée pour calculer la visibilité en environnement 2D et 3D vectoriels.

# Chapitre 1 Problématique, concepts et méthodes du domaine

## 1.1 Problématique d'ensemble

Google établit un lien interactif entre les images (ou la visibilité en sens large) et la carte. Autrement dit, si l'on dispose d'une photographie, il est possible d'utiliser ce système pour trouver la position de la prise de vue. De la même manière, si l'on a la position d'un site que l'on souhaite visiter, le système peut nous décrire l'environnement visible depuis ce lieu grâce à l'imagerie. La continuité des images et du son intégré dans la vidéo donne plus d'informations qu'une série discrète d'images et il existe plusieurs solutions pour géolocaliser des vidéos décrivant un site géographique. Google Maps permet aussi de géolocaliser des vidéos. Un simple clic sur un emplacement de la carte affiche la vidéo qui a été précédemment géolocalisée à cet endroit (Figure 2).

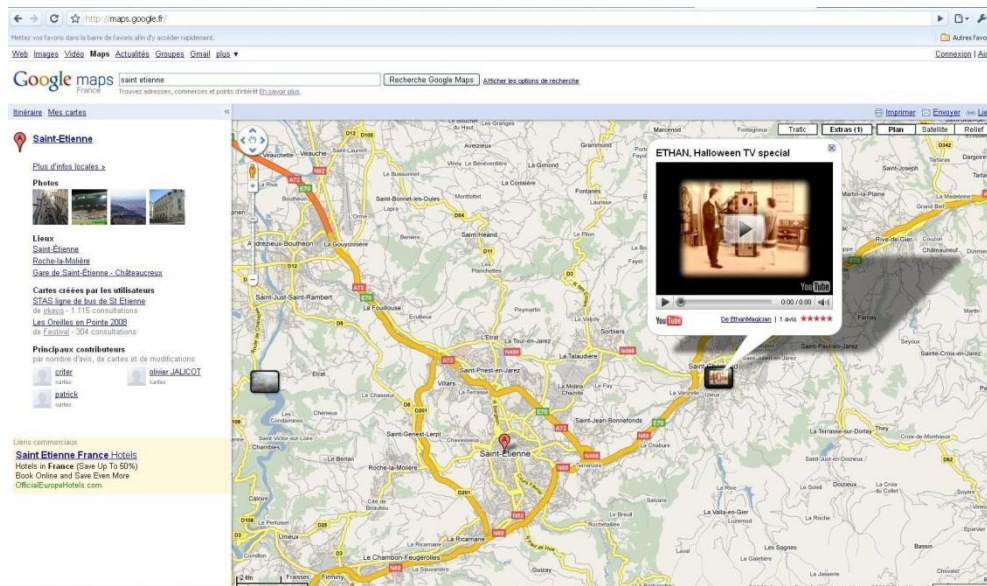


Figure 2. Géolocalisation de la vidéo sur l'internet

Geo-Movie est une application développée pour le sous-titrage géographique et temporel. Elle est utilisée dans les vidéos capturées par les sous-marins. Cet outil inscrit directement dans le corps de la vidéo des données telles que l'heure, la date, la longitude, la magnitude, la profondeur,... et les affiche en bas de l'écran (Figure 3).



Figure 3. Système GeoMovie<sup>1</sup>

RedHen système est une solution militaire qui a pour fonction de géolocaliser les informations photo et vidéo capturées par avion. Une interface permet d'afficher la vidéo en parallèle avec la carte qui fournit la position de la caméra (Figure 4).

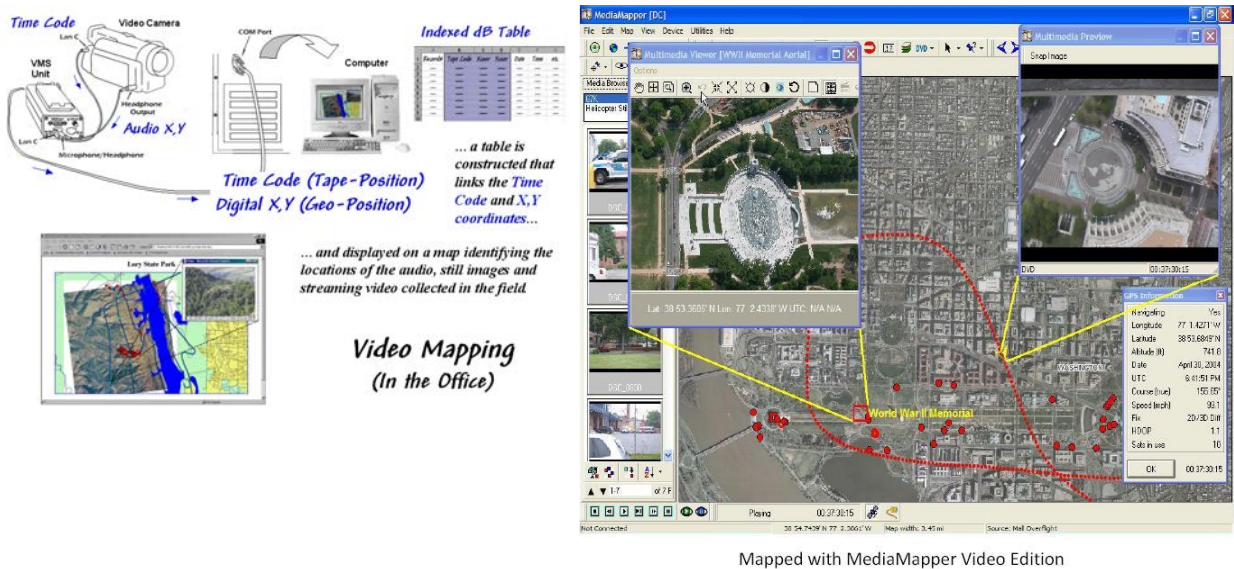


Figure 4 RedHenSysteme<sup>2</sup>

<sup>1</sup> www.justmagic.com

Acadia Video-GPS est un système militaire d'exploration qui utilise un SIG et un système de traitement d'images pour géolocaliser les cibles dans la vidéo, et organiser les missions sur le terrain (Figure 5).



Figure 5 Acadia Video-GPS<sup>3</sup>

On peut aussi envisager une approche différente, en assurant la géolocalisation non pas au moment de la prise de vue, mais ensuite, à partir du contenu de la photographie ou de la vidéo. On pense bien sûr à une géolocalisation par la reconnaissance de certains objets de la photographie en question dans d'autres photographies ou vidéos déjà localisées. Cela peut fonctionner essentiellement en milieu urbain et plus exceptionnellement en milieu rural pour des bâtiments ou monuments facilement reconnaissables. Cela apparaît difficile à envisager pour des lieux plus banals sans monuments facilement reconnaissables. En revanche, en milieu urbain, les bâtiments possèdent des façades remarquables dont la géométrie particulière peut permettre leur reconnaissance dans les photographies. Mais cela nécessite de gérer le problème complexe des multiples angles de vue d'un même lieu qui perturbe l'appariement de deux photographies prises dans des conditions de prise de vue différentes.

---

<sup>2</sup> [www.redhensystems.com](http://www.redhensystems.com)

<sup>3</sup> [www.tjmelectronics.com](http://www.tjmelectronics.com)



Saint-Etienne, cathédrale St Charles



Grotte Vallée de Cotatay (Le Chambon-Feugerolles)

Figure 6 La reconnaissance des objets présentés dans les images

Par ailleurs, plutôt que de partir au hasard à la recherche d'éléments d'images, il semble intéressant de s'appuyer sur les bases de données géographiques existantes et plus spécifiquement sur les modèles 3D qui reconstituent de manière plus ou moins fidèle un environnement urbain ou rural. Il est de plus en plus courant de mettre en correspondance des modèles numériques 3D des lieux avec des objets qui font l'objet d'une captation *in situ*. Cette association peut servir à superposer visuellement à ce qui est visible sur le terrain des objets numériques localisés. Il s'agit d'une approche de réalité augmentée qui ajoute aux scènes vues à travers un dispositif spécial des éléments tirés de l'environnement numérique en 3D. Il s'agit pour l'instant d'objets numériques simples dans les applications grand public, mais on utilise de plus en plus des environnements complexes, intégrant des bâtiments disparus ou à construire qui s'ajoutent à la vue réelle de l'utilisateur.

Une autre approche est de connecter, aux environnements 3D d'un lieu, des objets issus d'une captation *in situ*. Un des exemples les plus célèbres et les plus aboutis est la combinaison de Google Earth et de Google StreetView. Google StreetView localise des images prises à la volée, grâce à un véhicule spécialement conçu pour ce genre d'acquisition. Les photographies sont ensuite visibles dans Google Maps, sur fonds cartographiques, sous forme d'une série d'images panoramiques. Les photographies géolocalisées prises à 360 degrés sont mises en continuité avec la reconstitution numérique 3D de l'environnement (Figure 7-8).

## Analyse de visibilité et géolocalisation en milieu urbain avec un modèle numérique 3D



Figure 7. Environnement 3D dans Google Earth



Figure 8. Google Street View

Ce couplage entre vues *in situ* et modèle numérique ouvre un grand nombre d'applications dans de multiples domaines. Dans certains lieux, Google superpose aux vues 360° de Street



View dans Google Map les photographies prises par des amateurs et publiées sur le site Panoramio<sup>4</sup>. Un site comme <http://www.whatwasthere.com/> associe des photographies anciennes à des images de Street-View (Figure 9). Il devient alors possible de les visualiser en superposition aux images à 360°.

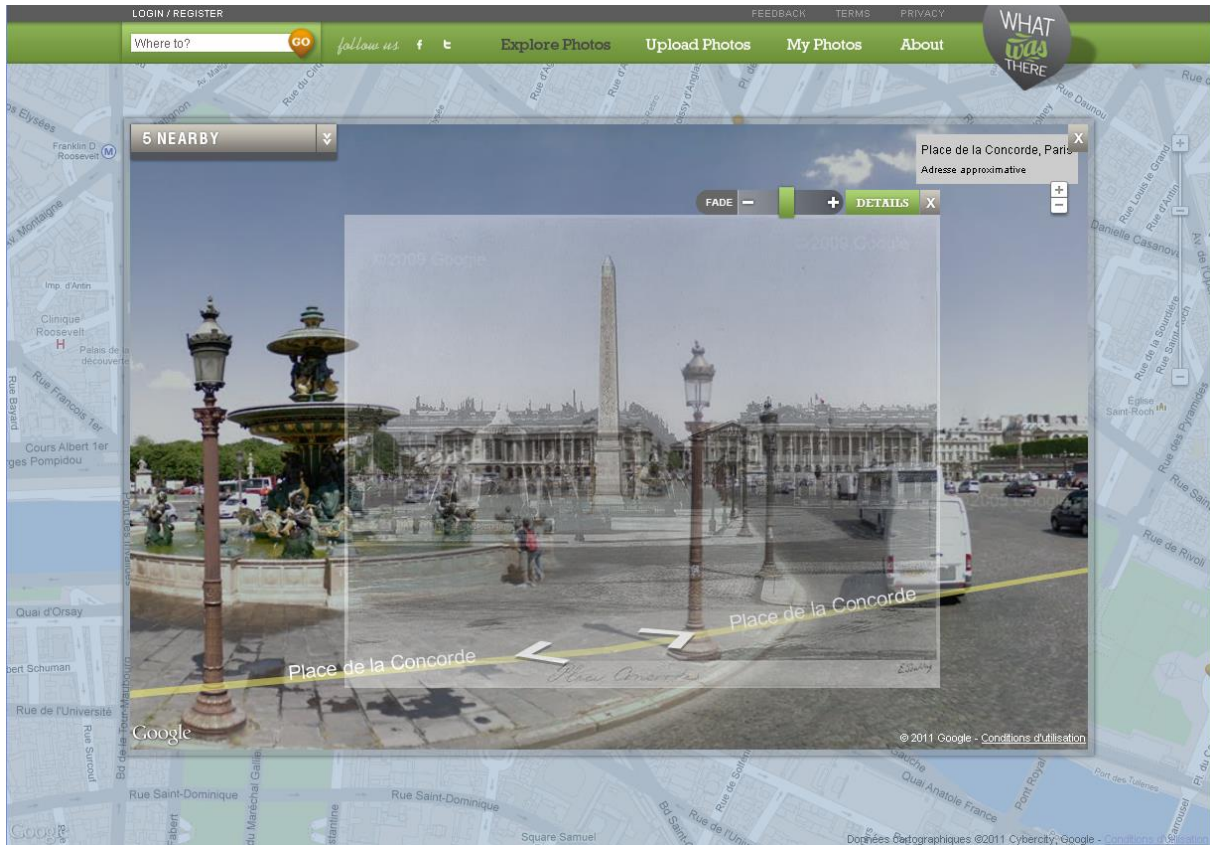


Figure 9. Association des photographies anciennes à des images de Street-View

Ce couplage photographie/modèle numérique 3D peut être utilisé pour géolocaliser la photographie. Dans le présent travail, nous proposons d'aller chercher dans la base de données des indicateurs préalablement stockés, qui permettraient d'associer automatiquement les images aux bâtiments. Si chaque façade d'un bâtiment dispose d'un indicateur de texture<sup>5</sup>, il est possible d'apparier une image avec un bâtiment à partir de cet indicateur et donc de localiser l'image. La généralisation des modèles numériques 3D géospécifiques urbains, dans lesquelles on habille sur chaque façade de bâtiment une image captée automatiquement sur le terrain, va faciliter la constitution de bases de données géographiques comportant un indicateur de texture.

<sup>4</sup> [www.panoramio.com](http://www.panoramio.com)

<sup>5</sup> La texture est liée à la présence de fenêtres, au changement des couleurs et aux éléments en relief de la façade.

La démarche inverse est aussi envisageable. Elle consiste à documenter les objets présents dans la photographie et la vidéo en fonction de ce qui est théoriquement visible depuis un point de vue donné. Il s'agit de reconstituer dans l'environnement numérique un équivalent de la captation effectuée dans le monde réel. Cela nécessite de pouvoir effectuer le calcul de l'espace visible, qu'on nomme dans la littérature isovist, fondé sur un modèle numérique de l'environnement. Ce calcul permettrait de décrire les objets que l'on peut théoriquement voir dans le modèle 3D. Il serait aussi possible de produire des indices de visibilité multiples associés aux objets.

Ces différents exemples amènent à concevoir une méthode complète qui analyse les objets de l'image pour la localiser, puis calcule la visibilité afin d'estimer les objets potentiellement visibles depuis la position initiale afin d'associer ces photographies ou vidéos aux objets de la base de données, et, réciproquement, d'associer aux photographies prises de ce point de vue des informations issues de la base de données. Dans ce cadre, ce travail de recherche porte spécifiquement sur deux problématiques : la localisation d'une prise de vue et l'analyse de visibilité en milieu urbain à l'aide d'un modèle 3D. Pour la localisation d'une prise de vue par image, une méthodologie a été développée afin d'enrichir une base de données urbaine SIG grâce à un descripteur de textures de façade, calculé sur des images de référence. Ce descripteur est ensuite utilisé pour retrouver ce bâtiment dans une nouvelle photographie non localisée, et le localiser dans une base de données SIG 3D afin d'estimer la position et l'orientation de la prise de vue dans le monde réel. La qualité des résultats obtenus est évaluée en fonction des paramètres de prise de vue de la photographie. La deuxième problématique porte sur le calcul de la visibilité en 3D, pour lequel nous présentons deux méthodes. La première utilise un nuage de points 3D et la seconde un nouvel algorithme basé sur l'angle solide<sup>6</sup>. La preuve de concept est établie dans un environnement SIG 3D des bâtiments, reconstitués numériquement et intégrant le relief. Notre étude est basée sur des technologies existantes en plein développement. Les résultats obtenus montrent qu'elles peuvent être mobilisées dans une grande variété d'applications.

---

<sup>6</sup> [http://fr.wikipedia.org/wiki/Angle\\_solide](http://fr.wikipedia.org/wiki/Angle_solide)

## **1.2 Reconnaissance des bâtiments dans une image et géolocalisation de la prise de vue**

La reconnaissance des éléments géographiques dans la littérature se limite actuellement aux bâtiments. La reconnaissance des bâtiments est une étape essentielle pour développer les usages de la réalité augmentée dans un environnement urbain. En effet, la réalité augmentée est un moyen efficace pour ajouter des annotations ou des objets virtuels sur les façades des bâtiments. Les informations ajoutées peuvent par exemple être historiques, si l'on se réfère à un bâtiment ancien, ou informationnelles comme la programmation d'un lieu culturel ou les horaires des trains dans une gare. Nous pouvons utiliser aussi cette identification comme un point de repère pour orienter des visiteurs ou des touristes. Des applications pour les portables ont déjà été imaginées par Amlacher, Paletta, Luley, Fritz et Almer (2008).

Notre problématique est d'estimer la précision des calculs par rapport à la réalité. Plusieurs critères sont à prendre en compte pour cette estimation :

- Notre modèle est-il complet ? Autrement dit, contient-il tous les éléments nécessaires : couvert végétal, infrastructures électriques ou publicitaires, nouveaux bâtiments, etc.
- Notre modèle est-il précis ? Les positions et les hauteurs des bâtiments ou des arbres sont-elles correctes ?
- Notre modèle est-il à jour ? Un arbre en hiver qui n'a pas de feuille n'aura pas la même influence sur la visibilité que le même arbre en été.

Une des raisons de la différence entre le calcul du champ de visibilité et la réalité est liée à la prise en compte de la portée de la distance de vue. Pour calculer le champ de vision nous définissons une distance ou un rayon de vision  $R$ . Cette distance est essentielle pour que le calcul soit faisable technologiquement. Cette limitation dans la distance de vision peut causer une différence avec la réalité. Par exemple si la distance du calcul de champ de visibilité est de 1 km nous ne distinguons évidemment pas un bâtiment ou une montagne situés à 5 km, pourtant parfaitement visibles dans la réalité.

Nous explorons deux méthodes de reconnaissance, le premier basé sur l'appariement entre l'image de synthèse et l'image réelle, le second sur la recherche de texture.

### **1.2.1 Reconnaissance des bâtiments par l'appariement entre l'image de synthèse et l'image réelle**

L'image réelle est l'image qu'on peut avoir par une prise de vue dans un espace quelconque. L'image de synthèse est l'image obtenue à partir d'une modélisation en 3D de cet espace. Dans la littérature, nous avons trouvé plusieurs tentatives de couplage entre des images réelles en 2D et les images de synthèses en 2D, pris dans des modèles SIG 3D, par appariement des primitives (coins, lignes) comme dans la Figure 10, cette opération est appelée couplage (2D/3D). Le couplage (2D/3D) est essentiel pour les applications de réalité augmentée, pour l'extraction de texture et pour les applications de navigation virtuelle. Microsoft et Shum (1998), Debevec et al (1996) et Karner et al (2002) demandent à l'utilisateur de faire ce couplage de manière manuelle. Teller et al (2003), Liu et Stamos (2005) proposent l'utilisation d'un modèle 3D LIDAR à haute résolution, encore cher et peu disponible. Reitmayr et Drummond (2006) utilisent un modèle SIG 3D texturé pour améliorer l'extraction des côtés de façades et le processus d'appariement. Sourimant (2009) proposent une méthode supervisée pour réaliser l'appariement. Cette méthode est basée sur l'algorithme de RANSAC écrit par Fischler et Bolles (1981). La méthode de Sourimant (2009) est construite à partir de trois éléments :

- 1) Estimation initiale de la position et l'orientation de la prise de vue qu'on appelle matrice de transformation<sup>7</sup>.
- 2) Existence d'un modèle 3D précis.
- 3) Possibilité d'extraire les bâtiments sans occlusion (voiture, être humain, autre bâtiment).

Bioret et al (2009) ont trouvé un moyen de calculer l'angle entre deux façades d'un bâtiment en utilisant les points de fuite présents sur l'image. Ils recherchent ensuite le même angle entre deux façades dans la base de données 2D des bâtiments, établissant ainsi un couplage entre l'imagerie et la base de données géographiques.

Dans la thèse qu'il a soutenue l'an dernier dans notre laboratoire, Youssef Attia propose de son côté une approche systématique de ce problème fondée sur l'appariement de lignes extraites de photographies d'ensembles de bâtiments urbains avec celles extraites d'un échantillon systématique d'images de synthèse produites à partir d'un modèle numérique 3D

---

<sup>7</sup> La matrice de transformation pour la caméra définit la relation entre les coordonnées des points 3D dans le monde réel et des points 2D sur l'image d'une prise de vue. On la nomme aussi matrice de projection ([http://fr.wikipedia.org/wiki/Calibration\\_de\\_caméra](http://fr.wikipedia.org/wiki/Calibration_de_caméra)).

(Attia 2012). Cette approche statistique est utilisable potentiellement pour localiser des photographies sans référencement géographique précis.



Figure 10. Le couplage entre l'image 2D et le modèle 3D SIG.

### 1.2.2 Reconnaissance des bâtiments à partir de leurs textures

Cette méthode est basée sur des images stockées de bâtiments. La méthode cherche à mettre en relation la texture des images stockées et la texture dans l'image de recherche. Cette méthode a été développée par Yu-Chia Chung, Han et Zhihai He (2009) Amlacher, Paletta, Luley, Fritz et Almer (2008) Zhang (2005). La mise en relation est faite par une extraction d'un descripteur de texture.

Dans la littérature il y a deux types de descripteurs de textures :

- Le descripteur global de texture que l'on trouve dans les travaux de Haralick, Shanmugam et Dinstein (1973) et Ojala, Pietikäinen et Mäenpää (2002). Ce genre de descripteur est utilisé pour la classification de textures par Khoo, Ong et Wong (2008) comme les texture des forêts, des bâtiments etc. Nous n'avons pas utilisé ces méthodes car notre objectif était de chercher spécifiquement à identifier les façades des bâtiments.
- Le descripteur local d'une texture qui est un ensemble de caractéristiques des pixels de l'image autour des points d'intérêt. Un point d'intérêt est un point possédant une valeur spéciale (maximum ou minimum local, par exemple) d'une fonction mathématique appliquée sur l'image. Un exemple de cette fonction est le filtre de Canny, qui détecte les contours dans une image. Le descripteur local d'une texture permet de trouver ces points d'intérêt dans une autre image du même objet en utilisant les mêmes caractéristiques locales.

Mikolajczyk et Schmid (2005) ont fait plusieurs tests sur les descripteurs locaux disponibles. Ils ont trouvé de bons résultats en utilisant la méthode SIFT (Scale-Invariant Feature Transform) par Lowe (2004). Cette méthode est robuste pour les changements d'échelle, les rotations et les transformations affines. Nous revenons plus précisément sur une évaluation de cette approche ci-dessous au § 2.2. Bay et al (2008) ont également proposé une méthode intéressante en utilisant un descripteur de texture SURF (Speeded Up Robust Feature). Ce descripteur permet des calculs plus rapides que le descripteur SIFT et il est déjà appliqué dans les applications pour appareils portables<sup>8</sup>. Dans les tests que nous avons effectués sur Matlab avec le code SIFT développé par VLFEAT1 et le code SURF développé par OPENSURF2, la méthode de SIFT a donné de meilleurs résultats dans le cas d'un changement d'angle de vue. C'est pour cette raison que nous avons utilisé la méthode SIFT.

Le problème des méthodes locales vient des erreurs dans les appariements. Pour reconnaître les façades d'immeubles dans une image, il faut d'abord extraire le descripteur SIFT et le comparer ensuite aux descripteurs SIFT stockés dans la base de données SIG. On choisit alors la façade qui a le plus grand nombre de points d'intérêt coïncidant avec les points d'intérêt de l'image. L'appariement, c'est-à-dire la mise en correspondance entre les points d'intérêt, est réalisé en fonction d'un seuil déterminé de distance entre les descripteurs locaux. Il peut parfois se produire des erreurs d'appariement en utilisant la méthode SIFT qui sont dues au fait que le descripteur ne décrit qu'une partie locale de l'image autour du point d'intérêt (Figure 11. ).

---

<sup>8</sup> <http://www.vision.ee.ethz.ch/~surf/download.html>

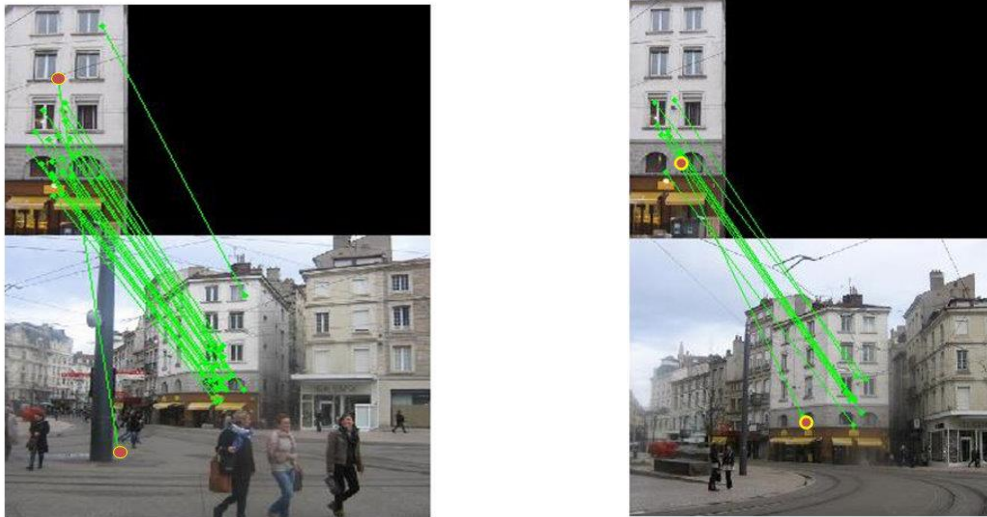


Figure 11. Descripteur SIFT. Erreurs d'appariement

Dans ce cas, il existe des méthodes pour éviter les faux appariements entre les façades et le sol dans des reconstructions d'environnement 2.5D ou 3D. Elles utilisent un environnement Manhattan-like (Olufs et Vincze 2011), (Jung et Kim 2012), ou les propriétés de forme, couleur, perspective et texture (Hoiem, Efros, et Hebert 2007). D'autres méthodes sont proposées par Xiaojuan et Wang (2013). Nous n'avons pas utilisé ces méthodes pour deux raisons principales : 1) elles sont limitées à un environnement disposant d'un sol plat, donc limitée pour rendre compte d'une situation géographique quelconque ; 2) même dans ce cas ces méthodes réclament un temps considérable de calcul. Pour répondre à ce défi, nous avons adapté une autre solution basée sur la géométrie de la façade.

Le grand avantage des méthodes de texture locale sur les méthodes d'appariement est qu'elles ont une meilleure performance en cas d'occlusion et que l'on n'a pas besoin d'une pré-estimation du positionnement, de l'orientation ou de la matrice de transformation de la prise de vue pour procéder à la reconnaissance des façades (voir la discussion générale dans le paragraphe 2.2).

### **1.2.3 La géolocalisation à partir des images**

L'utilisation d'images pour estimer le positionnement et l'orientation d'une prise de vue n'est pas nouvelle, en particulier en milieu fortement urbanisé. En effet, dans ce cas, l'erreur de positionnement du GPS couplé à l'appareil peut s'avérer importante à cause des problèmes de réflexion du signal et de la réception limitée des satellites comme le montrent les recherches de Li, Taylor, Kidner et Ware (2006) et Bioret, Moreau et Servières (2009). La

reconnaissance des bâtiments est largement utilisée dans le cadre de la navigation virtuelle pour améliorer l'aspect visuel des modèles 3D. La texture des bâtiments présents sur les photographies est aussi utilisée pour l'enrichissement de l'environnement virtuel par Sourimant, L. Morin, K. Bouatouch et De Rennes (2009). Les méthodes existant dans la littérature utilisent généralement une base des données géoréférencées de prises de vue. Par exemple Zhang et Kosecka (2006) proposent une méthode intéressante pour trouver la position de la prise de vue d'une image donnée par deux images géoréférencées en milieu urbain, en faisant l'hypothèse d'un bon chevauchement entre les trois images. Robertson et Cipolla (2004) font un placement approximatif de l'image et ils procèdent alors à une amélioration en limitant la recherche des images à proximité. Mais cette méthode ne calcule pas l'orientation de la prise de vue. Micusik et Kosecka (2009) utilisent la même technique sur une série d'images consécutives. Les auteurs définissent les références pour les deux premières images. Ils trouvent alors la position de la troisième photo en utilisant l'hypothèse d'un chevauchement constant entre trois images consécutives. Cette opération est appliquée de manière récursive pour géoréférencer la série des images. Johannson et Cipolla (2002) ont proposé de stocker une texture simplifiée des façades des bâtiments par extraction des côtés verticaux et horizontaux. Cette idée résout le problème du stockage des images entières. Snavely et al (2006) utilisent dans un premier temps une méthode de reconstruction 3D à partir de mouvement (Structure From Motion) afin de construire un modèle 3D sans échelle définie. Ensuite, ils alignent la projection verticale du modèle avec une image satellitaire géoréférencée par des opérations de similarité (échelle, rotation, translation). Mais une intervention manuelle est indispensable dans cet algorithme pour établir le positionnement et l'orientation de la prise de vue. Cette méthode fait l'hypothèse d'une connaissance a priori de la zone concernée pour effectuer le géoréférencement de l'image.

### **1.3 L'isovist**

Dans la littérature, le premier emploi du terme d'isovist revient à Benedict (1979) qui le définit comme l'espace visible à partir du point où se situe l'observateur supposé. Gibson (1983) et Conroy Dalton et Bafna (2003) le définissent, eux, comme l'agrégation des lignes de vue qui partent de l'observateur. On trouve en fait deux termes en anglais pour nommer cet espace visible : « isovist » et « viewshed », dont l'emploi varie en fonction des méthodologies et des applications. « L'isovist » est plus utilisé pour les calculs de visibilité dans un environnement urbain ou à l'intérieur d'un bâtiment, avec des données au format vectoriel.



Un « viewshed » désigne plutôt l'espace visible d'un observateur situé dans un environnement naturel et rural, le calcul étant fondé sur des données raster comme le rappellent Floriani et Magillo (2003).

La prise en compte de la troisième dimension dans le calcul de visibilité est un réel défi. Seules quelques solutions peuvent détecter les obstacles 3D qui limitent l'isovist. Le calcul automatique de l'isovist 3D pose en effet des problèmes spécifiques, en particulier quand le sol n'est pas plan. Dans la suite, nous allons explorer des méthodes différentes pour calculer le champ de vision dans un environnement de dimension 2 puis 3.

### 1.3.1 Calcul de l'isovist en 2 dimensions

Dans cette étude on va considérer que l'observateur est placé au centre d'un cercle de rayon  $R$ . Les objets dans ce cercle sont des polygones à  $N$  cotés. La Figure 12 présente la situation.

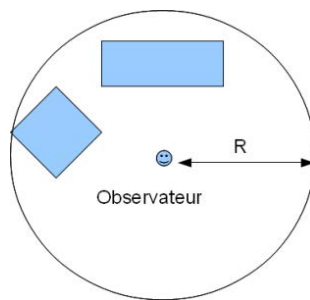


Figure 12. L'observateur et les objets dans un environnement à 2 dimensions

#### Le lancement ou tracé de rayons

Dans la littérature, la méthode la plus connue pour calculer l'isovist est le lancement ou tracé de rayons (« ray tracing »). Le principe est de lancer  $M$  rayons dans toutes les directions autour l'observateur, l'angle entre deux rayons consécutifs étant égal à  $360^\circ/M$ . Ensuite on cherche le point d'intersection le plus proche avec les objets de l'espace qui font obstacle au regard. L'ensemble de ces points définit le polygone de visibilité. La résolution de ce calcul dépend du nombre de rayons lancés. La Figure 13 illustre l'algorithme.

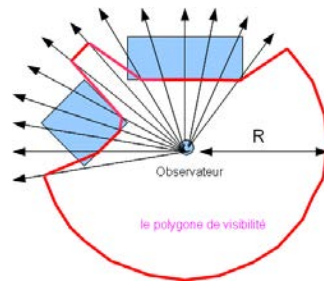


Figure 13. Le lancement des rayons

Cet algorithme est utilisé par Rana (2006) dans l'application qu'il a développée pour calculer l'isovist en 2 dimensions avec le logiciel Arcview 3 de la société ESRI. Pour obtenir une bonne précision, il utilise une résolution angulaire de  $0.1^\circ$  et procède donc à  $360/0.1 = 3600$  lancements de rayons depuis chaque point. Pour un environnement de  $N$  segments, la complexité de cet algorithme est d'ordre  $M*N$  intersections mutuelles entre les segments 2D. Le test d'intersection entre ces segments peut être réalisé en utilisant l'algorithme de la ligne de balayage « Sweep line algorithm » par Bentley et Ottmann (1979). La complexité de cet algorithme pour  $S$  segments est de l'ordre de  $S*\log(S)$ .

### Agent en déplacement (walking agent)

Une autre méthode a été proposée par Batty and Jiang (1999) et Batty (2001), celle de l'agent en marche (walking agent). Le principe est semblable au lancement des rayons. On imagine un observateur qui marche sur une ligne droite depuis un point donné avec un pas constant jusqu'à ce qu'il arrive à un obstacle. On répète cette opération dans toutes les directions à partir du point de départ. L'angle entre deux directions consécutives est déterminé *a priori* comme dans le cas du lancement des rayons. La méthode est présentée dans la Figure 14.

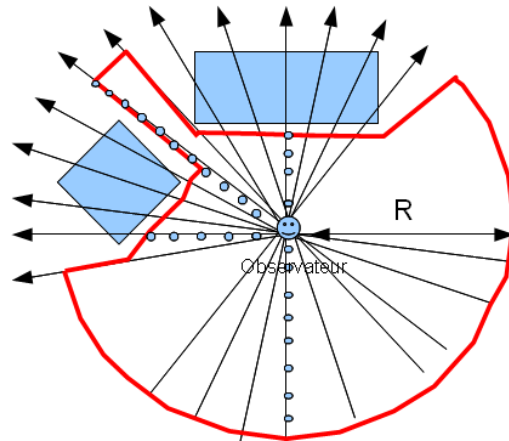


Figure 14. L'agent en déplacement

La complexité de cet algorithme dépend du nombre de directions de la marche et de la longueur du pas de l'agent. A chaque pas il faut tester si l'agent est entré en collision avec un objet (polygone) de l'environnement. Cette opération est équivalente à un test d'intersection avec tous les côtés ( $N$  segments) des polygones de l'environnement. Dans le cas précis, si l'agent marche sur une distance  $R$  dans  $M$  directions avec un pas  $d$ , on trouve en 2 dimensions  $M*R/d*N$  segments dont l'intersection est à tester.

### L'intervisibilité

La dernière méthode proposée est le calcul d'intervisibilité. Cette méthode a été proposée par Turner et al. (2001). L'idée est de générer un maillage régulier de points dans l'espace. Ensuite on teste pour chaque point du maillage l'intervisibilité avec l'observateur. Cela consiste à vérifier si la ligne entre le point de l'observateur et le point de maillage a ou non une intersection avec les objets de l'espace. La complexité de cette méthode dépend du nombre de points dans le maillage. Si nous supposons que le nombre de points est  $k$ , alors l'ordre de la complexité en dimension 2 pour cette méthode est  $k*N$  tests d'intersection mutuelle entre les segments. La Figure 15 présente la méthode de l'intervisibilité.

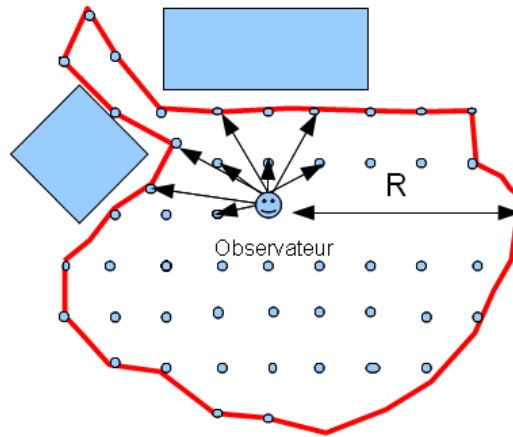


Figure 15. L'intervisibilité

### 1.3.2 Les programmes de calcul du champ de la visibilité en dimension 2

Une liste des programmes pour calculer le champs de vision en dimension 2 est présentée par Do (1997). Ces programmes utilisent des techniques différentes comme le lancement des rayons (Tcl-Light), le « shadow casting », la suppression des murs occlus dans l'environnement d'AutoCAD avec AutoLisp (Do 1994) et la recherche des murs visibles (Do 1995). Christenson (2010) a calculé la perméabilité visuelle dans l'architecture en utilisant l'espace visible. D'autres logiciels sont disponibles en ligne<sup>9</sup>.

### 1.3.3 Les méthodologies de calcul du champ de la visibilité en dimension 3

L'idée la plus courante est d'utiliser un MNT (Modèle Numérique de Terrain) en mode raster (nom donné au mode image en géomatique) et de calculer la visibilité par le lancement de rayons. Plusieurs solutions ont été proposées par Franklin et Ray (1994), Floriani et Magillo (1994), Van Kreveld (1996), Lake, Lovett, Bateman, et Day (2000), Floriani et Magillo (2003), Brossard, Joly et Tourneux (2008), Fishman, Haverkort et Toma (2009). La Figure 16 explique l'idée du calcul de la visibilité en mode raster.

<sup>9</sup> <http://www.visibility.org/>

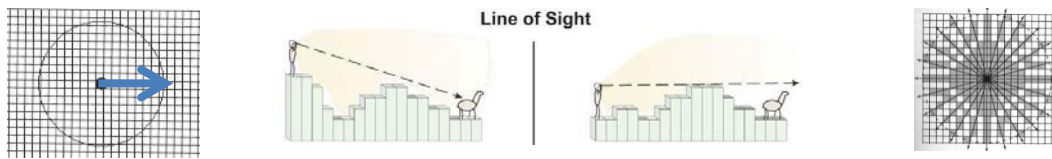


Figure 16. La visibilité en mode raster, les points sombres sont visible et les points clairs ne sont pas visibles. Source (Brossard, Joly et Tourneux 2008)

Pour calculer la visibilité, on teste l'intervisibilité entre le pixel de l'observateur et chaque pixel de l'image raster. Comme nous pouvons le voir, la précision du calcul dépend de la résolution de l'image raster, mais l'augmentation de celle-ci accroît la complexité du calcul. Pour améliorer le résultat du calcul sans augmenter sa complexité du même ordre de grandeur, une approche multi-résolution est possible. On utilise alors une forte résolution à proximité du point d'observation jusqu'à une distance donnée et une résolution plus faible pour le reste de l'image raster. La Figure 17 illustre ce principe.

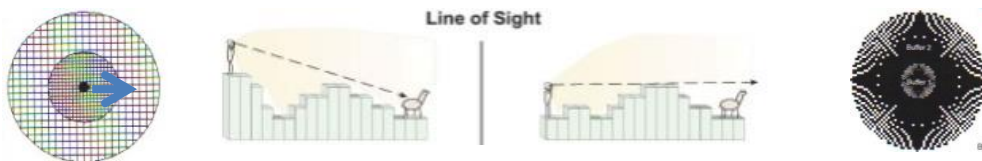


Figure 17. La visibilité en mode raster avec plusieurs résolutions, les points sombres sont visible et les points clairs ne sont pas visibles.

Source : (Brossard, Joly et Tourneux 2008).

Dans cette approche, l'intégration des éléments du sursol tels que les bâtiments ou la végétation se fait par extrusion en ajoutant à l'altitude des pixels du MNT la hauteur de ces éléments. Cette technique est utilisée par exemple dans les logiciels Arcgis<sup>10</sup> et SkylineGlobe<sup>11</sup>.

Le problème de cette méthode est que le modèle de données ne permet pas d'intégrer dans le calcul de visibilité les faces verticales des éléments comme les façades des bâtiments par exemple. Une solution pour le traitement de l'élévation verticale des bâtiments est suggérée par Bartie et al. (2008) (2010). Il s'agit de découper les façades en cellules régulières selon une valeur de hauteur donnée et de procéder au calcul de visibilité pour chaque cellule d'une manière analogue aux pixels en mode raster (Figure 18).

<sup>10</sup> <http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//009z000000v8000000.htm>

<sup>11</sup> <http://www.skylinesoft.com>

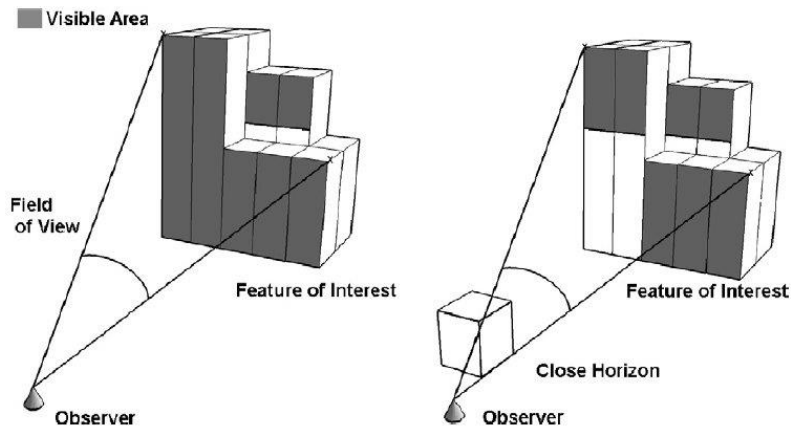


Figure 18. La visibilité en mode raster avec des façades. Source (Bartie et al. 2010)

Une autre méthode consiste en l'emploi d'une pixellisation volumique de type « voxel », élément volumique élémentaire d'un maillage tridimensionnel qui permet d'intégrer des éléments comme des bâtiments ou un couvert végétal. On peut aussi associer aux voxels un coefficient de transparence en fonction de la nature des objets (voir Pyysalo, Oksanen, et Sarjakoski (2009)). Dans la Figure 19 les bâtiments sont intégrés avec des voxels opaques par Morello et Ratti (2009). Ce type de méthode nécessite de disposer de données 3D Lidar, encore assez coûteuses et qui ne sont pas disponibles partout.

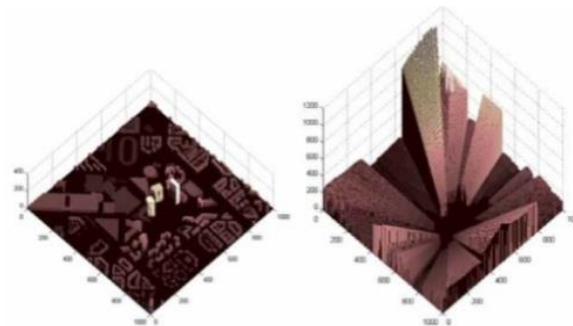


Figure 19 Modèle voxel avec des bâtiments (Morello and Ratti 2009)

Une utilisation d'un modèle vectoriel a été proposée par Bilsen (2009). Celui-ci a développé un programme de calcul de visibilité en 3 dimensions qui utilise le lancement de rayons dans un environnement urbain avec un terrain plat (Figure 20).

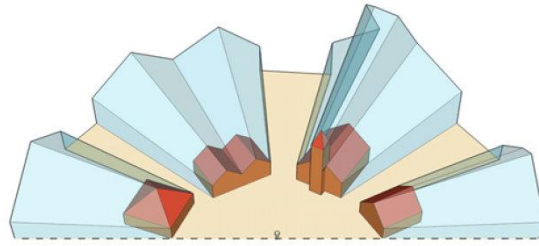


Figure 20 Le champ de visibilité en 3D avec un terrain plat (Bilsen 2009)

La précision et la complexité sont les mêmes que pour les calculs en deux dimensions et dépendent du nombre de rayons lancés.

D'autres méthodes de calcul de visibilité ont été proposées plus récemment par Yin et Carswell (2012). Mais il n'existe pas à notre connaissance de solution pour un calcul de visibilité avec des données vectorielles en 3 dimensions sur un terrain avec une topographie quelconque.

#### **1.3.4 La syntaxe spatiale (space syntax)**

Dans la littérature on trouve de nombreux essais d'analyse de la relation entre les formes bâties et la perception humaine. Ittelson (1961) a étudié la relation entre les activités humaines et l'environnement urbain pour étudier la perception visuelle de l'espace. Ashihara (1984) montre que selon des observations, les gens s'arrêtent habituellement un instant au point d'intersection de deux routes par reflexe, exemple d'une relation simple et directe entre une forme et une pratique de l'espace. Lynch (1976) a utilisé le champ de vision pour analyser les effets visuels des obstacles et de la configuration du terrain. Hillier et Hanson (1984) ont présenté la syntaxe spatiale (space syntax) comme un outil pour analyser les caractères visuel de l'espace urbain et pour prédire le comportement humain suivant ces caractères. La syntaxe spatiale produit un groupe d'indices tels que la visibilité, la fermeture, l'ouverture ou l'échelle qui sont fondés sur le calcul du champ de visibilité. Putra et Yang (2005) et Putra (2005) ont trouvé une corrélation entre ces indices et la perception visuelle humaine de l'espace. Fisher-Gewirtzman et al (2005) ont défini l'indice de l'espace ouvert (Space Open Index) comme l'espace libre (sans obstacles) que nous pouvons avoir à partir d'un point et d'une distance de vision donnés. Bilsen (2010) a proposé, pour la modélisation d'un site de travail, des indices de visibilité dans un environnement 3D vectoriel.

Bilsen a calculé des statistiques sur la distance aux obstacles de l'environnement (minimum, maximum ou déviation standard) sur plusieurs plan horizontaux. Ces indices aident à

l'évaluation de la sécurité de déplacement des personnes dans le site. Il a calculé encore l'angle solide<sup>12</sup> pour le ciel visible, qui évalue la sensation d'enfermement et le degré d'aération sur un lieu de travail.

## 1.4 Conclusion

Comme nous l'avons vu, peu de recherches dans le domaine de la géolocalisation d'une prise de vue utilisent la géométrie du contenu de l'image. Dans cette thèse, nous présentons une méthodologie pour l'enrichissement d'une base de données urbaine SIG grâce à un descripteur de texture de façade calculé sur des images de référence. Cet indicateur est ensuite utilisé pour retrouver ce bâtiment dans une nouvelle image, le localiser dans une base de données SIG 3D, et estimer sa position et son orientation dans le repère de l'appareil photographique qui a pris le cliché. La qualité des résultats obtenus fait l'objet d'une discussion. La prise en compte de la troisième dimension dans le calcul de visibilité est aussi un réel défi. Seules quelques solutions peuvent détecter les obstacles en 3D qui limitent l'isovist. Dans cette thèse, nous présentons un nouvel algorithme qui peut détecter tous les objets qui bloquent la vue dans un environnement 3D reconstitué numériquement et intégrant le relief. Une démonstration avec des données SIG est également effectuée.

---

<sup>12</sup> [http://fr.wikipedia.org/wiki/Angle\\_solide](http://fr.wikipedia.org/wiki/Angle_solide)





# **Chapitre 2 Reconnaissance de bâtiments et localisation de photographies au moyen d'un descripteur de texture**

La reconnaissance automatique des bâtiments est une étape essentielle pour la réalité augmentée et un outil possible de la géolocalisation d'une prise de vue. Les recherches dans ce domaine n'utilisent pas la géométrie du contenu de l'image. Ce chapitre présente une méthodologie pour l'enrichissement d'une base de données urbaine SIG grâce à un descripteur de texture de façade calculé sur des images de référence. Cet indicateur est ensuite utilisé pour retrouver ce bâtiment dans une nouvelle image et le localiser dans une base de données SIG 3D, afin d'estimer sa position et son orientation dans le repère de l'appareil photographique qui a pris le cliché. La qualité des résultats obtenus fait l'objet d'une discussion. Le contenu de ce chapitre a fait l'objet d'une publication en 2012 dans la Revue Internationale de Géomatique, vol. 22/3, pp. 439-459.

## **2.1 L'enrichissement de la base des données**

Dans ce chapitre, nous faisons l'hypothèse que l'on travaille avec une base de données existante des images des façades. Les photos sont prises avec un protocole bien défini : axe de vision verticale au plan de façade, distance de moins de 100 mètres et résolution 800\*600 pixels comme dans la Figure 21.



Figure 21. Des exemples d'images découpées correspondant aux façades des bâtiments.

Comme nous n'avons pas à notre disposition de base de données de ce type, il a fallu la construire spécialement. Pour ce faire, il existe deux méthodes : une méthode manuelle longue et fastidieuse et une méthode automatique fondée sur des images couplées avec leur modèle 3D SIG (Sourimant, L. Morin et K. Bouatouch 2009). Pour l'heure, la méthode automatique est limitée aux bâtiments simples et isolés. Comme nos bâtiments ne sont pas nécessairement isolés, et afin de contrôler la qualité et les conditions de prise de vue des photographies, nous avons utilisé la méthode manuelle.

A partir de ces images, on extrait des indices de description de texture qui sont associés aux polygones correspondant aux façades des bâtiments dans la base de données géographiques. On utilise pour ce faire un descripteur local de texture. Le descripteur local d'une texture est un ensemble de caractéristiques des pixels de l'image autour de points d'intérêt. Comme on l'a déjà évoqué dans le § 1.2.2 on a utilisé la méthode SIFT (Scale-Invariant Feature Transform) par Lowe (2004) (Figure 22).

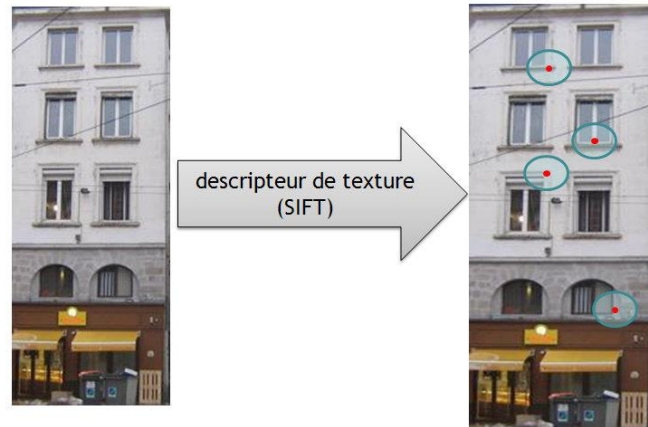


Figure 22 Extraction des quelques points d'intérêt remarquables de l'image de façade par analyse de texture

Après avoir extrait les points d'intérêt de la texture, on détermine leur positionnement selon les trois dimensions  $(x,y,z)$ . Cette étape est réalisée en appliquant des contraintes d'homographie sur le plan formé par la façade du bâtiment. Les contraintes d'homographie sont exprimées par une matrice qui établit la correspondance entre deux images du plan point à point par Hartley et Zisserman (2004). Grâce à cette matrice, il est possible de déterminer le positionnement des points d'intérêt dans le monde réel (Figure 23). Pour calculer cette matrice d'homographie, nous devons connaître le positionnement des quatre coins de la façade sur le modèle 3D et les dimensions de l'image de façade.

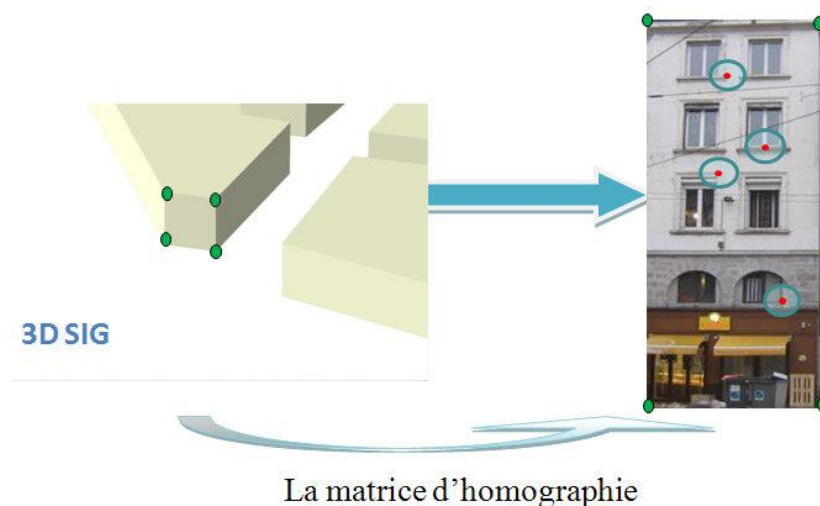


Figure 23. *Trouver la position  $(x,y,z)$  des points d'intérêt*

Ainsi, à la fin de ce processus, le descripteur de texture sera composé de 4 paramètres:

- La position des points d'intérêt suivant l'algorithme SIFT.
- Les descripteurs locaux de ces points suivant l'algorithme SIFT.
- La dimension de l'image de référence.
- Les positions (x,y,z) de ces points d'intérêt dans le monde réel.

## 2.2 La reconnaissance des façades des bâtiments dans une photographie quelconque

Pour reconnaître les façades d'immeubles dans une image, il faut d'abord extraire le descripteur SIFT de l'image et le comparer ensuite avec les descripteurs SIFT stockés dans la base de données SIG. Plusieurs travaux existent dans ce domaine, voir Picard, Cord et Valle (2009), A. Le\_Bris et N. Papanastasiou (2010). On cherche les façades qui ont un nombre minimal de points d'intérêt coïncidant avec les points d'intérêt de l'image (ce nombre est fixé à 8, valeur qui est apparue adéquate dans notre expérimentation). L'appariement, c'est-à-dire la mise en correspondance des points d'intérêts, est réalisé en fonction d'un seuil déterminé de distance entre leurs descripteurs locaux. Il peut parfois se produire des erreurs d'appariement en utilisant la méthode SIFT qui sont dues au fait que le descripteur ne décrit qu'une partie locale de l'image autour du point d'intérêt (Figure 24).

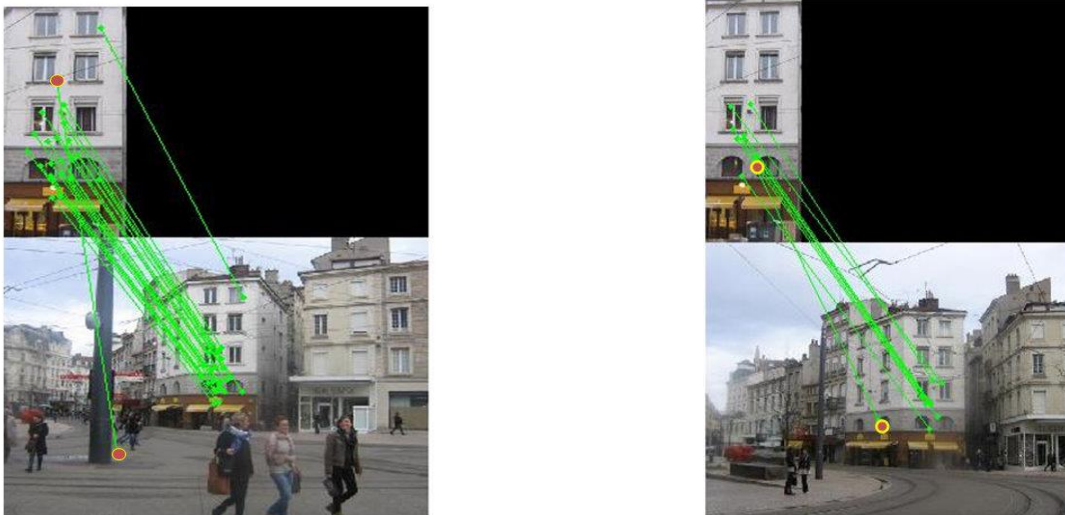


Figure 24. *Descripteur SIFT. Erreurs d'appariement*

Une manière de résoudre ce problème serait d'adapter manuellement le seuil d'appariement à chaque cas. Nous proposons comme solution d'appliquer la contrainte homographique sur les points d'appariement, puisque les appariements se font entre les images de la même façade. De plus, cette contrainte contrôle la distribution géométrique des points d'intérêt sur l'image de façade. Cette idée permet de conserver le plus grand nombre d'appariements et elle élimine toutes les erreurs. Le calcul de la matrice d'homographie est fondé sur les méthodes Direct Linear Transformation et RANSAC par Fischler et Bolles (1981) dans lesquelles la matrice d'homographie a besoin de quatre appariements pour être calculée. Le résultat obtenu est ensuite amélioré par la méthode d'itération non-linéaire basée sur la minimisation de la distance de Sampson par Hartley et Zisserman (2004). Une fois ce calcul effectué, et après avoir éliminé les faux appariements de l'algorithme SIFT, on considère que la façade de test est détectée si le nombre d'appariements est supérieur à 8, ce nombre ayant donné des bons résultats sur la reconnaissance des façades. Sinon on considère que la façade de test ne se trouve pas dans la photographie. Le système sélectionne la façade qui a le plus grand nombre d'appariements entre les façades détectées et la présente comme le résultat le plus pertinent de l'identification. Cette façade unique sera suffisante pour estimer la position de la prise de vue dans l'étape suivante. De la matrice d'homographie calculée et de la dimension de l'image de référence stockée, on déduit les quatre coins de la façade dans la nouvelle image. En résultat, on a identifié la façade dans l'image et on l'a localisée dans la base de données SIG 3D (Figure 25).

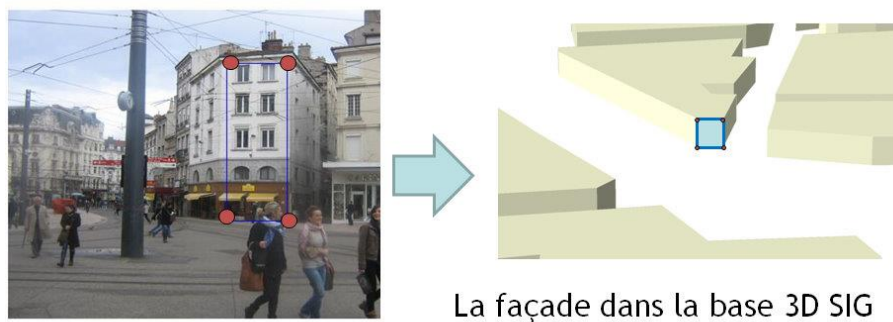


Figure 25. La reconnaissance de façade dans l'image et la géolocalisation dans une SIG 3D

Bien sûr, si la photographie comprend plusieurs façades, le système peut aussi détecter les autres façades présentes (Figure 26). Leur identification doit alors être cohérente avec la position géographique de la façade dominante et avec les nombres de points d'appariement.

Dans notre test les façades situées à plus de 20 mètres de la façade dominante identifiée ont été éliminées. Il faut noter que dans la plupart des cas on ne détecte qu'une seule façade dans l'immeuble, vraisemblablement à cause de l'angle que forment les autres façades par rapport à l'angle de prise de vue ou parce qu'elles ne sont pas complètes.

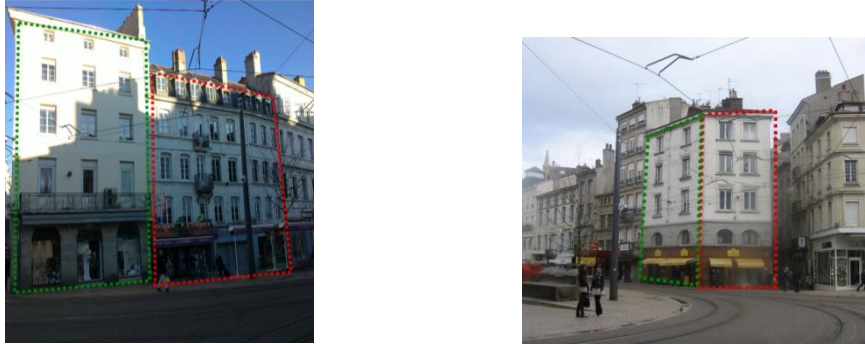


Figure 26. La reconnaissance de plusieurs façades dans la photographie

60 tests ont été effectués en jouant sur différents paramètres pour valider la méthode sur notre base des données: angle, éclairage, occlusion, etc.. Les images de test sont présentées dans la (Figure 27). Le nombre d'appariements et donc la performance de la méthode dépendent en effet de plusieurs facteurs:

- le degré de complétude de la vue : un manque de recul dans une rue étroite occasionnera une façade incomplète dans l'image prise. Si l'image contient moins de 75% de la surface de la façade, cela peut provoquer une fausse identification.
- l'occlusion : L'occlusion liée à la présence d'objets au premier plan conduit à une réduction du nombre d'appariements. C'est le défi le plus important dans notre approche car des objets, en particulier mobiles (piétons, voitures...) qui sont situés sur les photographies, peuvent cacher des détails qui font la différence entre deux façades similaires. Cette occlusion pose alors un problème d'identification. D'après nos tests il semble possible avec la méthode proposée, d'identifier une façade avec une occlusion allant jusqu'à 25% de la surface de l'image.
- la rotation de l'image : théoriquement, le descripteur de texture SIFT est invariable par rapport à la rotation, mais le nombre d'appariements diminue pourtant avec la rotation. Le nombre d'appariements dans notre jeu test s'est avéré cependant suffisant pour identifier le bâtiment, même après rotation.

- la distance à la façade : l'éloignement de la prise de vue par rapport à la façade réduit la résolution de l'image de la façade, ce qui a pour conséquence de diminuer le nombre d'appariements. Dans notre test, on parvient à identifier une façade jusqu'à une distance de 200 m.
- le changement d'éclairage : le changement d'éclairage réduit aussi le nombre d'appariements. Les photographies de notre base de données ont été prises vers midi. Pourtant, on parvient à retrouver ces façades dans des photos prises à toute heure de la journée. Des problèmes d'identification se posent toutefois en cas d'ombre portée d'un autre bâtiment sur la façade. Dans le cas de photos prises en soirée, la qualité des résultats baisse fortement car l'éclairage public ou l'éclairage du bâtiment modifient de façon trop importante l'apparence de la façade.
- l'angle de vue : le changement d'angle de vue réduit le nombre de mises en correspondance. En effet, la méthode SIFT n'est pas invariante pour les transformations affines (Lowe 2004). Dans nos tests, l'identification est possible jusqu'à 40° de changement par rapport à l'image de la base de donnée, prise avec un axe de vision perpendiculaire au plan de la façade. Le résultat est donc très satisfaisant pour ce paramètre.
- la résolution d'image : l'augmentation de la résolution des photographies augmente le nombre de points d'intérêt et améliore donc le résultat de la reconnaissance puisque l'on dispose d'un plus grand nombre de points d'appariement. En revanche, l'application de la méthode SIFT sera plus difficile avec une image à grande résolution, car on a alors besoin d'une capacité de calcul plus importante. Finalement, ce paramètre dépend du type d'application de la méthode. Si le traitement se fait à partir d'un appareil portable, il faut choisir une résolution basse. Si la photo est traitée par un serveur offline, on peut utiliser une résolution plus grande. Dans notre cas on a utilisé une image de petite dimension (600\*800 pixels), sur laquelle les traitements sont rapides.
- la dimension de la façade : si la façade est de grande taille, elle aura d'une manière générale un plus grand nombre de points d'intérêt. Cette façade sera alors identifiable à grande distance. Dans nos tests, les bâtiments ont des dimensions allant jusqu'à 30 mètres de hauteur et jusqu'à 50 mètres de largeur.
- les changements sur la façade : si les couleurs de l'immeuble restent les mêmes entre deux ravalements, des différences de détail changent plus souvent : les devantures des magasins, les rideaux aux fenêtres ... Tout changement sur la façade réduit le nombre des mises en



correspondances. Dans nos tests, il est possible d'identifier la façade si les changements concernent moins de 25% de sa surface.

– la texturation de la façade : si la façade est bien texturée, elle aura plus de points d'intérêts, ce qui permettra une identification plus robuste.

Notre méthode trouve des limites dans deux cas principaux:

– les façades identiques : si deux façades sont parfaitement identiques, tant dans leur texture que dans leur dimension, leur identification n'est évidemment pas possible. Dans ce cas on élimine ces deux façades de la base de données afin de forcer le système à chercher d'autres façades pouvant être considérées comme la façade principale de l'image. Toutefois, si les dimensions des immeubles ayant un même type de façade sont différentes, la distribution de leurs points d'intérêt sur la façade est différente et permet de les distinguer. Il faut aussi prendre en compte les variations de texture entre deux bâtiments identiques dues à la différence des magasins en rez-de-chaussée.

– les façades en verre : une façade en verre reflète l'image des autres bâtiments et du ciel suivant l'angle de vue. Ce genre de façade n'est pas identifiable par notre méthode car les effets de réflexion sont complexes et varient en fonction des conditions atmosphériques.

Plusieurs images de test sont présentées dans la Figure 27.

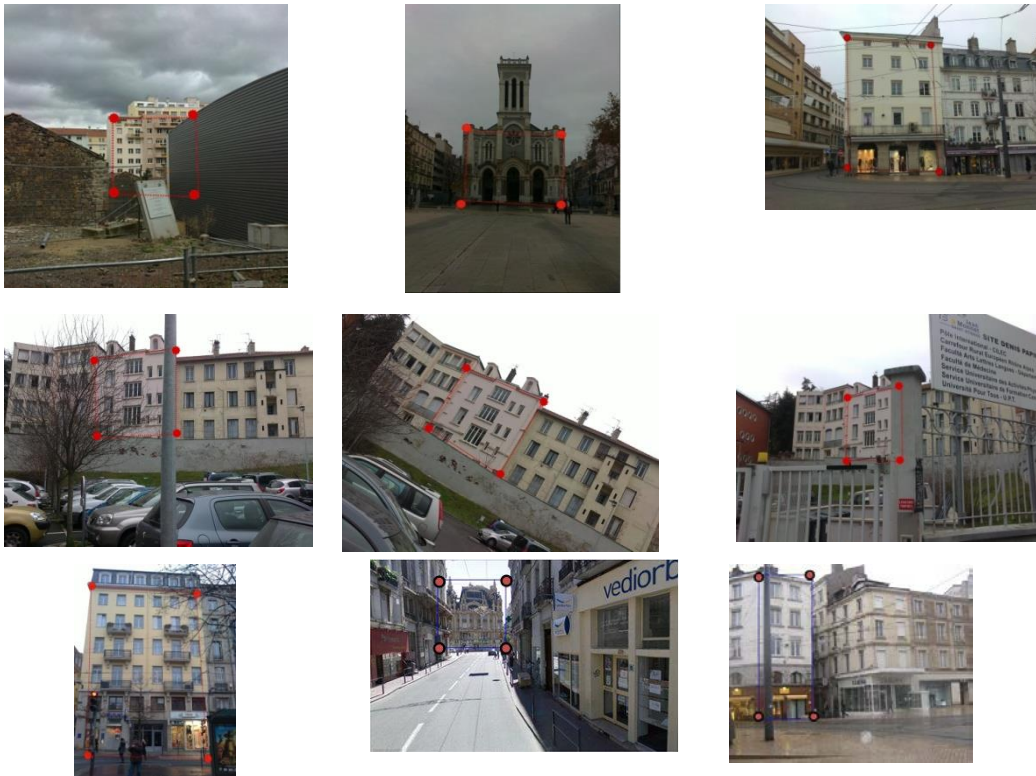


Figure 27. Plusieurs tests de reconnaissance des façades suivant des conditions de rotation, d'éclairage, de point de vue et d'occlusion différents, Les façades sont bien détectées avec des erreurs parfois dans le tracé des limites.

## 2.3 La géolocalisation d'une prise de vue

Après la reconnaissance d'une façade dans l'image et dès lors que l'on possède les paramètres intrinsèques de la prise de vue, on peut estimer la géolocalisation de cette dernière. En effet, une façade est un plan. Il s'agit donc de trouver la matrice de transformation qui permettra d'obtenir la position et l'orientation de la prise de vue à partir des points coplanaires. Le calcul de la matrice de transformation est implémenté dans plusieurs logiciels avec des méthodes bien explicitées<sup>13</sup>. Puisque notre descripteur associe à chaque point d'intérêt son positionnement 3D, les points d'intérêt dans notre image ont les mêmes coordonnées 3D que les points d'intérêt du descripteur. On obtient ainsi un ensemble de points précisément définis par leur positionnement 2D dans l'image et 3D dans le monde réel.

<sup>13</sup> [http://www.vision.caltech.edu/bouguetj/calib\\_doc/htmls/example5.html](http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example5.html)

Dans la littérature, on trouve de nombreuses recherches pour estimer la matrice de transformation à partir de l'ensemble des points définis sur l'image 2D et de leur position 3D. Pour trois points non-colinéaires Holt et Netravali (1991), Wolfe et al.(1991) et Haralick et al. (1994) ont trouvé quatre solutions possibles. Pour quatre points sans triplets colinéaires dans l'image, Haralick (1989), Abidi et Chandra (1995), Yuan (1989) et Yang et al. (2009) montrent l'existence d'une solution unique de la matrice de transformation. Ces méthodes donnent un résultat plus précis si les quatre points de calcul sont bien distribués dans l'image. C'est pour cela que nous avons choisi pour notre méthode les quatre points définis par les quatre coins de la façade dans l'image.

Nous appliquons la méthode de Yang et al. (2009). Ensuite le résultat est amélioré en utilisant la méthode itérative non-linéaire de Levenberg-Marquardt (Marquardt 1963). Cet algorithme permet de diminuer l'erreur entre les points d'intérêt de l'image et la projection de leur position 3D, cette projection étant calculée suivant la matrice de transformation initiale de la méthode de Yang et al. (2009).

La qualité du résultat final est bien corrélée au nombre des correspondances réalisées entre les points d'intérêt de l'image et les points d'intérêt stockés pour la façade détectée. Tous les facteurs qui définissent le nombre des correspondances définissent aussi la précision de la géolocalisation de la prise de vue.

Pour notre test nous avons sélectionné des points de repère sur le terrain, puis nous avons utilisé les données du SIG complétées par l'image satellite de l'application Google Maps pour calculer la distance entre le résultat obtenu et la position réelle de la prise de vue. Avec nos 60 tests on constate que dans le cas où la prise de vue se situe à moins de 100 mètres de la façade détectée si la direction de l'axe de la vision est quasiment perpendiculaire au plan de cette façade, l'erreur du positionnement est de moins de 3 mètres et l'erreur d'orientation est de moins de 15°.

Analyse de visibilité et géolocalisation en milieu urbain avec un modèle numérique 3D



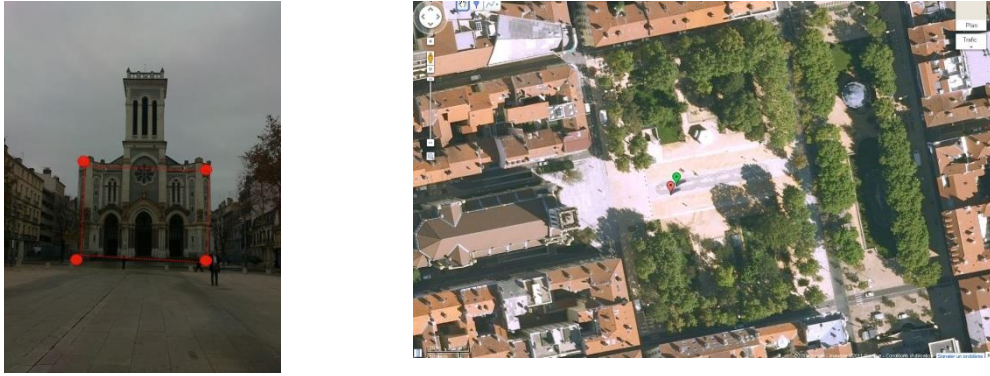


Figure 28. La géolocalisation d'une prise de vue à partir de l'identification d'une seule façade. Sur la carte, le point vert présente la position réelle et le point rouge présente la position mesurée. La position réelle est déterminée en fonction de points de repère sur le terrain.

La recherche d'une autre façade dans l'image est également possible. Si la nouvelle façade trouvée est coplanaire avec la façade initiale, on applique la même méthodologie en considérant les deux façades comme une seule façade. Le résultat est amélioré grâce à une distribution mieux répartie des points d'intérêt par rapport au calcul initial comme dans le travail de Yang, Cao, Lo et Zhang (2009) (Figure 29).



Figure 29 La géolocalisation d'une prise de vue à partir de l'identification de deux façades coplanaires. Dans la carte, le point vert présente la position réelle et le point rouge présente la position mesurée. La position réelle est déterminée en fonction de points de repère sur le terrain.

Si la nouvelle façade n'est pas coplanaire avec la façade initiale, la méthode de Yang et al. (2009) n'est plus applicable. On utilise alors l'algorithme SOFTPOSIT de David et al. (2004). Cet algorithme est basé sur quatre points qui ne sont pas coplanaires. Le résultat sera amélioré en appliquant la méthode itérative non-linéaire de Levenberg-Marquardt (1963).

Malheureusement le résultat obtenu dans ce cas est moins performant. En effet la nouvelle façade ajoutée présente généralement un angle de vue important (plus de  $30^\circ$ ) par rapport à l'image de la base de données, prise perpendiculairement au plan de la façade. Ce nouvel angle de vue important pose alors un problème du positionnement des points d'intérêt. Ceci provient toujours du fait que la méthode SIFT n'est pas invariante pour les transformations affines (Figure 30).



Figure 30 La géolocalisation d'une prise de vue à partir de l'identification de deux façades non-coplanaires. Dans la carte le point vert présente la position réelle et le point rouge la position mesurée. La position réelle est choisie selon des points de repère sur le terrain.

Comme il n'existe pas à notre connaissance de travaux analogues dans le domaine géomatique, et en général peu de recherches appuyées sur des données collectées directement dans le milieu urbain, il est difficile de comparer nos résultats à d'autres. Notre recherche reste à notre connaissance originale et ouvre des pistes d'application et de développement intéressantes.

## **2.4 Conclusion et pistes pour des travaux futur**

Dans ce chapitre, nous avons proposé et testé une méthode permettant d'identifier une façade d'immeuble présente dans une image quelconque et de la localiser dans une base de données SIG 3D. On peut dès lors déterminer l'endroit de la prise de vue à partir de la localisation de la façade détectée dans l'image, si les paramètres internes de la prise de vue, par exemple la

focale, sont connus. Cela suppose que la base de données SIG des bâtiments ait été préalablement enrichie par un descripteur de la texture des façades.

Le présent travail n'est qu'une première étape mais nos tests montrent des résultats pertinents et robustes. L'identification des façades semble possible avec des distances variables et des conditions de rotation, d'occlusion et d'éclairage différentes. L'erreur de positionnement commise est inférieure à 3 mètres en placement et inférieure à 15° sur l'orientation de la prise de vue. Ces résultats sont obtenus à condition qu'une des façades détectées soit quasiment perpendiculaire à la prise de vue, et située à moins de 100 mètres de celle-ci. Ceci n'est pas une réelle contrainte dans un milieu urbain. Une campagne complémentaire de tests doit cependant être réalisée pour confirmer et préciser les performances de la méthode.

La question se pose de la performance de la solution proposée en situation réelle. Il n'est pas envisageable de tester avec cette méthode toute la base de textures des bâtiments d'une ville pour chaque photographie. La proposition d'une solution générale dans le cas de grandes bases de données constitue une prochaine étape de la recherche. La voie la plus simple est de tenter de limiter *a priori* le nombre de textures candidates. Une première solution est de réduire le champ de recherche en s'appuyant sur une localisation approximative de la photographie. Dans le cas d'une photographie automatiquement géolocalisée par les antennes de téléphone portable, Wifi ou par GPS, on obtient une position avec une précision de 50 à 100 mètres en milieu urbain, ce qui permet de réduire drastiquement le nombre de bâtiments candidats. Dans le cas de photographies sans référence géographiques, ce serait à l'utilisateur de proposer une zone probable de localisation pour circonscrire la recherche. Une autre piste pour sélectionner les candidats est d'utiliser un indice de similarité rapide entre la photographie et les images de la base de données, tel que le *deep learning* qui a été présenté par plusieurs articles comme ceux de Bengio (2009) et Schmidhuber (2012). Cette idée est appliquée par Google<sup>14</sup> et a donné des résultats prometteurs.

Des tests en vraie grandeur, tout comme la généralisation de la méthode, nécessitent la constitution préalable d'une large base de données de signatures de façades pouvant servir à identifier les photographies. Plusieurs solutions systématiques existent pour constituer une base de données de ce type. De nombreuses villes se lancent dans la constitution de modèles 3D texturés détaillés, coûteux et complexes, fondés sur des systèmes de capture d'image combinant des techniques de stéréophotogrammétrie et de laser. D'autres systèmes fondés sur des images panoramiques sont développés par des entreprises pour servir de support à des

---

<sup>14</sup> <http://www.wired.com/wiredenterprise/2013/05/neuro-artificial-intelligence/all/>

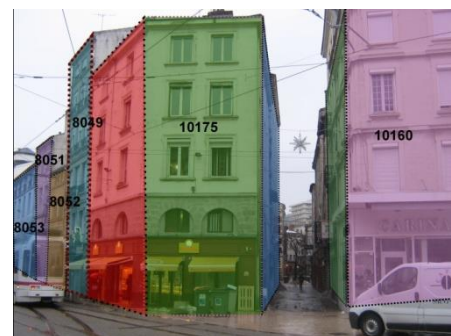
services de navigation potentiellement financés par la publicité. Google a Street View, Microsoft a Bing StreetSide et le groupe PagesJaunes a UrbanDrive qui utilise la technologie EarthMine. Ce type de services offre des opportunités importantes pour l'extraction de textures. Si les images panoramiques semblent offrir un intérêt, il faut là aussi développer des tests systématiques. Un premier essai avec des images Google ou EarthMine nous a montré que la résolution restait un peu faible pour permettre des appariements corrects. Par ailleurs, des problèmes de droit d'usage se posent pour un usage massif de données. Une solution de crowdsourcing fondée sur la fourniture d'images géoréférencées à l'adresse et dont des signatures de textures pourraient être extraites et stockées dans la base de données est aussi envisageable.

Les usages de la méthode proposée sont multiples. Elle peut faciliter la gestion des bases de données photographiques urbaines en aidant le gestionnaire à mieux localiser des photographies non encore géoréférencées. Il est aussi possible d'utiliser la méthode pour géolocaliser une prise de vue dans un endroit couvert où il n'y a pas de signal GPS, voire de venir en source d'information complémentaire des systèmes de navigation en milieu urbain où les GPS ont des problèmes de précision.

Ce mode de géolocalisation d'une prise de vue peut aussi constituer la phase initiale d'une démarche de couplage 2D/3D entre des images et une base de données géographique 3D comme celle proposée par Sourimant et al. (2009). Enfin, cette application nous permet d'identifier tous les bâtiments dans une image donnée. Autrement dit, à partir d'une identification d'une façade par notre méthode (Figure 31.a), on peut identifier les autres façades par le couplage 2D/3D (Figure 31.b).



(a) Identification de façade par le descripteur de texture, le numéro sur la façade et le numéro de bâtiment dans la base de données SIG 3D



(b) Couplage (2D/3D) suivant à la géolocalisation de la prise de vue par l'identification en (a)

Figure 31. Le couplage (2D/3D) basé sur la géolocalisation de la prise de vue



Comme on l'a vu, la performance du descripteur dépend de l'angle de vue de la façade car il est non invariant pour les transformations affines. Une perspective de poursuite de ce travail pourrait être d'utiliser un descripteur invariant pour les transformations affines tel que celui proposé par Mikolajczyk and Schmid (2004). Enfin, la méthode proposée pourrait être portée sur un appareil mobile. Il faut noter à ce propos que la méthode SURF donne de meilleurs résultats que la méthode SIFT, et que des applications de la méthode SURF existent déjà sur les appareils mobiles<sup>15</sup>.

On notera en conclusion que la performance de la méthode est meilleure si des bâtiments voisins sont dissemblables. Elle est donc plus particulièrement adaptée aux quartiers historiques des villes et aux centres des villages, en général dans les zones où les formes urbaines sont caractérisées par une grande diversité dans les types et les styles des bâtiments et la présence d'architectures spécifiques et originales. Les résultats risquent évidemment d'être moins bons dans les zones où l'architecture des bâtiments est plus uniforme comme dans les périphéries pavillonnaires ou les quartiers de tours et de barres.

---

<sup>15</sup> <http://www.kooaba.com/>

# Chapitre 3 Analyse de la visibilité urbaine par nuages des points 3D

Dans le premier chapitre, nous avons vu qu'il était difficile de réaliser un calcul de visibilité pour un environnement urbain établi sur un sol avec du relief. Les méthodes existantes se sont limitées à répondre à quelques questions : trouver les façades visibles à partir d'un point donné, trouver l'obstacle qui bloque l'intervisibilité entre deux points ou trouver l'effet sur l'intervisibilité de la construction d'un bâtiment dans un endroit quelconque. Dans ce chapitre nous proposons une méthode pour résoudre ce problème en utilisant un modèle SIG 3D vectoriel au moyen d'un maillage des points. Le contenu de ce chapitre a été accepté pour une communication à GISRUK, en 2011 : *3D Urban Visibility Analysis with Vector GIS Data. GISRUK, 27-29th April 2011, University of Portsmouth, UK.*

## 3.1 Environnement 3D vectoriel

L'idée principale de notre approche est de considérer l'espace comme une agrégation de polygones 3D. Pour le terrain nous utilisons un modèle TIN (Triangulated Irregular Network). Le modèle TIN est produit au moyen d'une triangulation de Delaunay à partir d'un maillage des points 3D (Yan et Lianhe 2009) (Figure 32).



Figure 32. Triangulation de Delaunay

Suivant ce modèle, chaque triangle a une position 3D bien définie. Pour le modèle de bâtiments, nous utilisons la méthode la plus connue qui est l’empreinte arienne des bâtiments avec extrusion. Dans ce genre de modèle nous pouvons redresser les façades et les toits des bâtiments comme des polygones 3D, qui ont un offset variant en fonction de la position du bâtiment sur le modèle TIN.

La Figure 33 présente une extraction 3D d’une partie de la ville de Saint-Etienne qui sert à notre expérimentation.

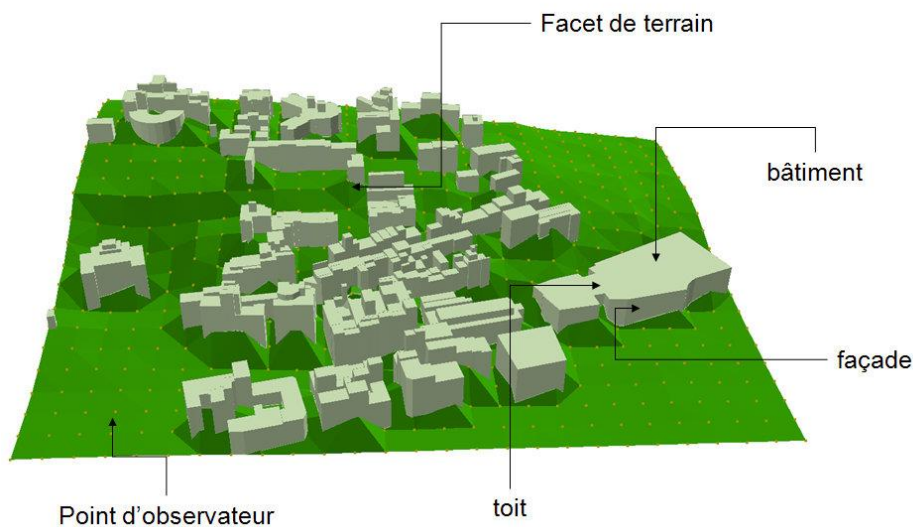


Figure 33. *Environnement 3D de la ville de saint Etienne*

Pour les tests de notre méthode, nous utilisons Matlab<sup>16</sup> comme environnement de développement avec l’extension de cartographie<sup>17</sup>. Nous avons choisi cette environnement logiciel parce qu’il peut importer et exporter des données géographiques et qu’il offre des moyens de modélisation, des fonctions mathématiques<sup>18</sup>, des opérateurs géomatiques en 3D<sup>19</sup>, des outils pour la visualisation en 3D<sup>20</sup> et une interface de débogage bien développée. Toutes ces fonctionnalités ne sont pas disponibles dans les outils SIG ordinaires.

<sup>16</sup> <http://www.mathworks.fr/>

<sup>17</sup> <http://www.mathworks.fr/products/mapping/>

<sup>18</sup> <http://www.mathworks.fr/help/techdoc/math/br81y5b-1.html>

<sup>19</sup> <http://www.cgal.org/>, <http://puddle.mit.edu/~glenn/kirill/SAGAHTML/saga.html>

<sup>20</sup> <http://www.mathworks.fr/help/techdoc/visualize/bqliccy.html>

## 3.2 Maillage des points et calcul d'intervisibilité

Pour le calcul de la visibilité nous avons besoin d'un maillage des points en 3D. Ce maillage des points peut être défini suivant les besoins de l'utilisateur. La génération du maillage est réalisée de manière automatisée par le programme, à partir d'un fichier 'shape', format standard de la société ESRI très utilisé dans le domaine des SIG, ou de manière manuelle par l'utilisateur. Pour commencer, nous pouvons utiliser les points 3D que nous avons utilisés pour générer le TIN avec une hauteur d'élévation (Figure 34).

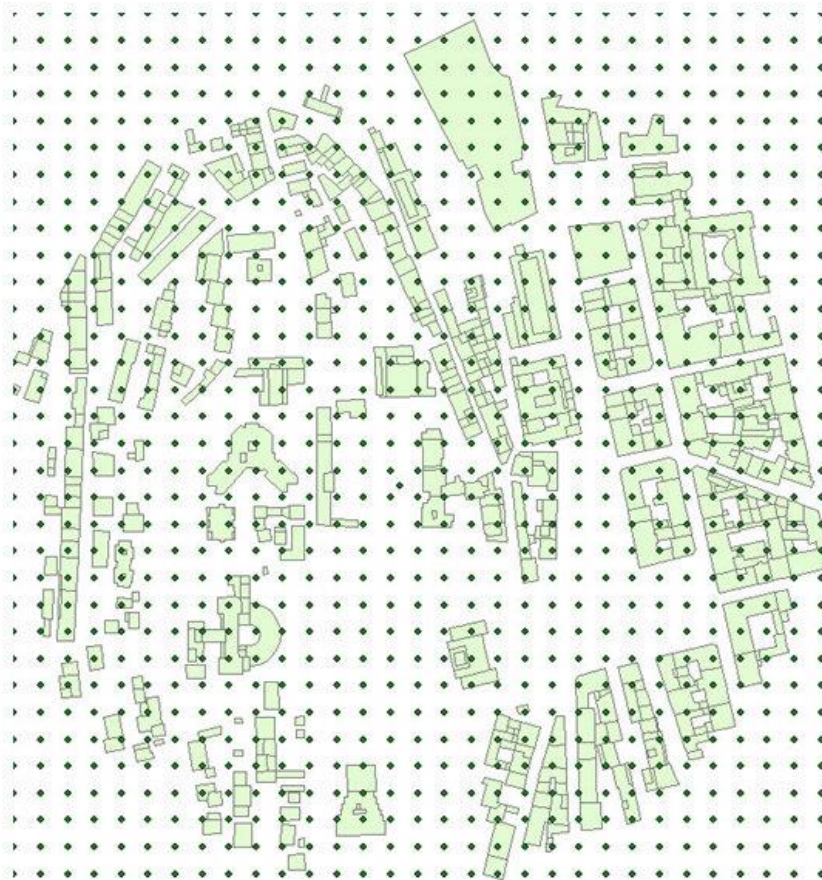


Figure 34. Le Maillage des points 3D suivant les besoins de l'utilisateur pour le calcul de visibilité

Ensuite pour calculer le champ de vision à partir du point d'observation, nous calculons l'intervisibilité entre l'observateur et les points du maillage. La valeur de l'intervisibilité pour le point de maillage 3D est égale à 1 si le segment 3D entre le point d'observation et le point du maillage n'est pas intersecté par un polygone 3D de l'espace (façade, terrain, toit). Nous

pouvons constater que l'opération d'intervisibilité est réciproque. Si le point *A* peut voir le point *B*, alors le point *B* voit le point *A*. Une interpolation entre les points du maillage suivant la valeur de l'intervisibilité est possible pour reconstituer une continuité visuelle.

La complexité de l'algorithme est de l'ordre de (nombre des points 3D de maillage) \*(nombre des polygones 3D de l'espace) opérations d'intersection entre un segment et un polygone en 3D. La précision de la méthode dépend du nombre de points du maillage.

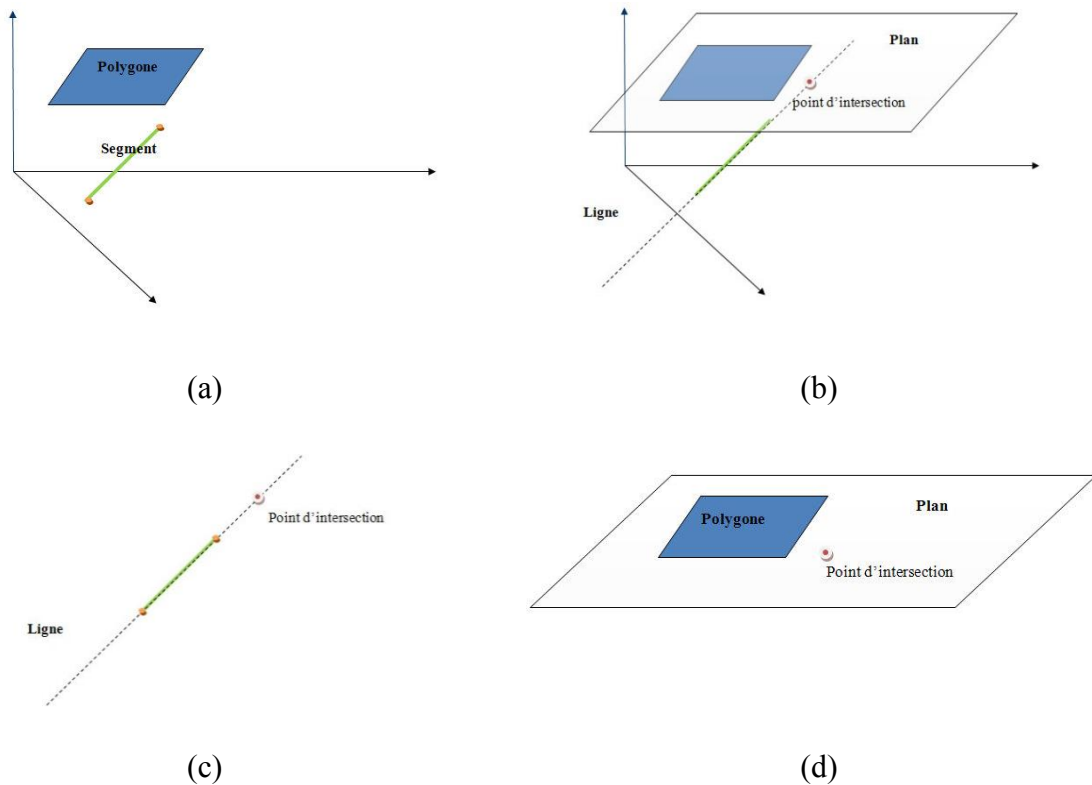


Figure 35. L'intersection entre un segment et un polygone dans un environnement 3D

L'algorithme pour calculer l'intersection entre un segment et un polygone en 3D suit cette méthode :

**Input (Polygone P, Segment S)**

Intersection\_Test = faux

**SI** le point d'intersection entre la ligne de segment et le plan du polygone dans l'environnement 3D (Figure 35b) **EXISTE**.

**SI** le point d'intersection est entre les extrémités de segment dans l'environnement 1D (Figure 35c).

**Si** le point d'intersection est dans le polygone dans l'environnement 2D (Figure 35d).

Intersection\_Test = Vrai

**Fin SI**

**FIN SI**

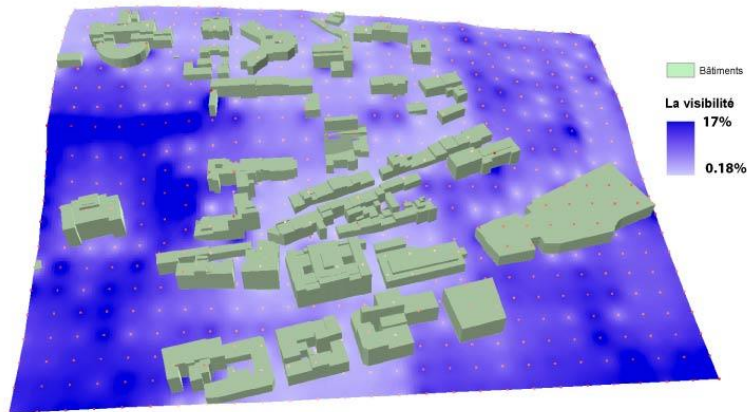
**FIN SI**

**Output (Intersection\_Test)**

Dans l'essai présenté dans ce document, la taille de chaque point du maillage est placée à 1,60 mètre au-dessus du niveau du sol, ce qui correspond à un individu humain de taille moyenne qui regarde de manière horizontale. Cette condition peut être modifiée pour répondre aux besoins d'une application spécifique. Par exemple, les points observés pourraient être à 0 mètre si l'observateur est un humain qui regarde un objet situé au sol, ou à 2.5 mètres si la personne regarde une affiche publicitaire par exemple. Dans ce calcul, un pourcentage de la visibilité ou un coefficient de visibilité peut être assigné à chaque point du maillage (Turner et al 2001) avec identification des points qui pourraient être vus du point considéré (Figure 36). Le résultat est exporté comme un fichier 'shape' des points, lisible par un logiciels SIG.

Attributes of VisGrIde							
FID	Shape	POINTID	GRID_CODE	Z	Relative_ZGH	VispPercentH	VisPointsIDC
12	Point	8763	5855273	585,6273	0,1	9,090909	, 8701, 8702, 8762, 8763, 8764, 8824, 8825, 8826
13	Point	8764	5839039	584,0039	0,1	10,227273	, 8701, 8702, 8703, 8763, 8764, 8765, 8825, 8826, 8827
14	Point	8765	5808027	580,9027	0,1	6,818182	, 8703, 8704, 8764, 8765, 8826, 8827
15	Point	8766	5730625	580,2825	7,22	7,954545	, 8703, 8766, 9012, 9013, 9073, 9074, 9134
16	Point	8767	5657601	565,8601	0,1	3,409091	, 8705, 8706, 8767
17	Point	8768	5596243	566,5243	6,9	2,272727	, 8705, 8768
18	Point	8769	5524016	564,3116	11,91	2,272727	, 8705, 8769
19	Point	8770	5421332	542,2332	0,1	7,954545	, 8707, 8708, 8709, 8710, 8770, 8771, 8832
20	Point	8771	5348356	534,9356	0,1	9,090909	, 8707, 8708, 8709, 8710, 8770, 8771, 8772, 8832
21	Point	8772	5271387	527,2387	0,1	4,545455	, 8707, 8710, 8771, 8772
22	Point	8824	5878066	587,9066	0,1	12,500000	, 8762, 8763, 8824, 8825, 8886, 8887, 8948, 9009, 9010, 9131, 9132
23	Point	8825	5867062	586,8062	0,1	13,636364	, 8762, 8763, 8764, 8824, 8825, 8826, 8886, 8887, 8948, 9009, 9070, 9131
24	Point	8826	5849551	585,0551	0,1	13,636364	, 8703, 8763, 8764, 8765, 8825, 8826, 8827, 8887, 8893, 8949, 9009, 9070

La carte d'inter-visibilité



La carte de l'inter-visibilité

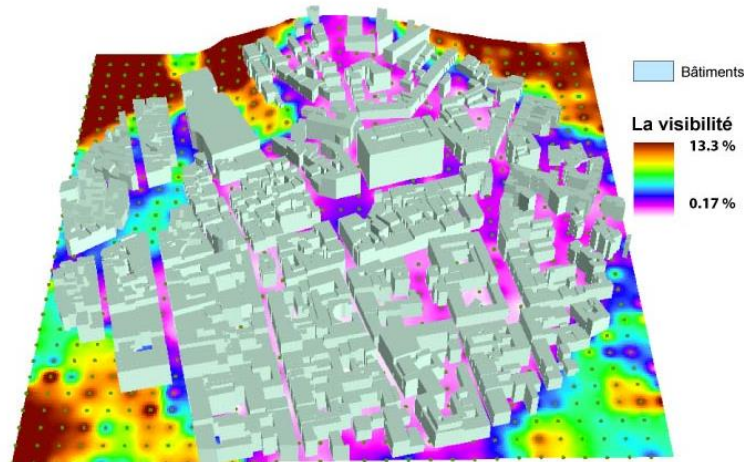


Figure 36 Carte interpolée par visibilité réciproque en utilisant la pondération inverse de distance (IDW). Dans la table, VisPersnetN est le pourcentage es points visibles, VisPointsIDC correspond aux IDs des points visibles depuis le point considéré.

Grâce à une option du programme, l'utilisateur peut ajuster le rôle des points du maillage qui sont situés dans les polygones de bâtiments. Il peut considérer que ces points n'interfèrent pas avec la visibilité ou il peut les considérer comme des points situés sur le toit d'un bâtiment. Dans ce cas ils affecteront le calcul de visibilité.

Les valeurs de visibilité des points à la frontière du maillage sont impactées par l'effet de bord. Les points en question pourraient être fortement visibles mais depuis des points situés en dehors de la région. Le maillage des points peut être prolongé afin d'ajouter des points sur les façades des bâtiments, les points du maillage pouvant être vus ou voyants ou les deux en même temps. La Figure 37 présente l'intervisibilité d'un maillage avec des points sur les façades.

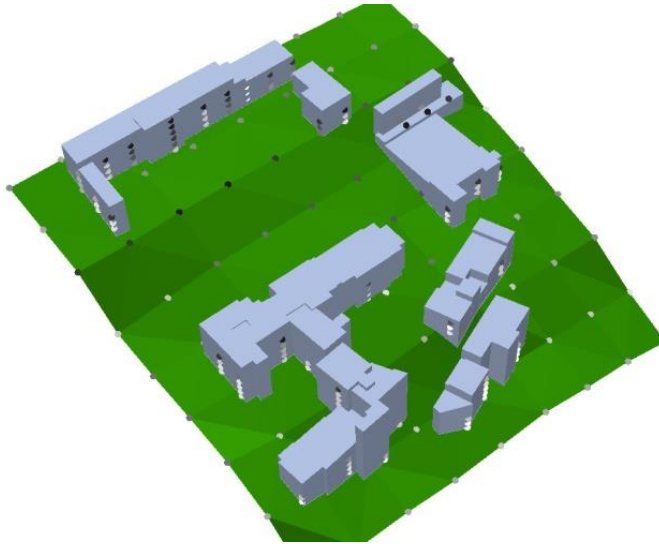


Figure 37. Le maillage de l'intervisibilité des points sur les façades, les points foncés ont une visibilité élevée et les points clairs ont une visibilité basse.

### 3.3 Calcul du champ de la visibilité

Le champ de visibilité peut être défini comme l'ensemble des points visibles dans l'environnement depuis le point d'observation. La méthode donne la possibilité d'associer à chaque point du maillage les obstacles qui bloquent la visibilité du point d'observation. Pour réaliser le calcul, nous devons identifier le point d'observation, sa hauteur au-dessus du niveau du sol, et identifier les points du maillage (position et la taille). La génération du maillage est réalisée de manière automatique par le programme, à partir d'un fichier shape ou de manière manuelle par l'utilisateur. Le résultat est un fichier shape des points avec une zone d'information de visibilité qui indique si le point du maillage est visible ou pas. Une table d'information des obstacles faisant écran à la vue énumère les zones de sol ou les bâtiments qui bloquent la vue de l'observateur (Figure 38).



Attributes of Viewshed1N									
FID	Shape	POINTID	GRID_CODE	Z	Relative_ZGII	Visp	VisObstacleC	ticket	
0	Point	7964	5309847	578,0918	1,6	1	None	Observer	
1	Point	7964	5309847	531,0847	0,1	0	0	Grid Point	
2	Point	7965	5306215	530,7215	0,1	0	0	Grid Point	
3	Point	7966	5307379	530,8379	0,1	0	0	Grid Point	
4	Point	7967	5314883	531,5883	0,1	0	0	Grid Point	
5	Point	7968	5321136	532,2136	0,1	0	0	Grid Point	
6	Point	7969	5309609	531,0609	0,1	0	0	Grid Point	
7	Point	7970	5344594	534,5594	0,1	0	0	Grid Point	
8	Point	7971	5349844	535,0844	0,1	0	0	Grid Point	
9	Point	7972	5351879	535,2879	0,1	0	0	Grid Point	
10	Point	7973	5362887	536,3887	0,1	0	0	Grid Point	
11	Point	7974	5379203	538,0203	0,1	0	0	Grid Point	
12	Point	7975	5375387	537,6387	0,1	0	0	Grid Point	
13	Point	7976	5357000	535,8	0,1	0	2152, 0	Grid Point	
14	Point	7977	5312101	531,3101	0,1	0	2152, 0	Grid Point	
15	Point	7978	5261367	526,2367	0,1	0	2152, 0	Grid Point	
16	Point	7979	5249379	525,0379	0,1	0	2152, 4448, 0	Grid Point	
17	Point	7980	5238930	523,993	0,1	0	2055, 2152, 4448, 0	Grid Point	
18	Point	7981	5230964	523,1964	0,1	0	2055, 2141, 2142, 2143, 2152, 4448, 0	Grid Point	
19	Point	7982	5220090	522,109	0,1	0	2093, 2143, 2144, 4448, 0	Grid Point	
20	Point	7983	5213461	521,4461	0,1	0	2093, 2094, 2144, 2151, 4448, 0	Grid Point	

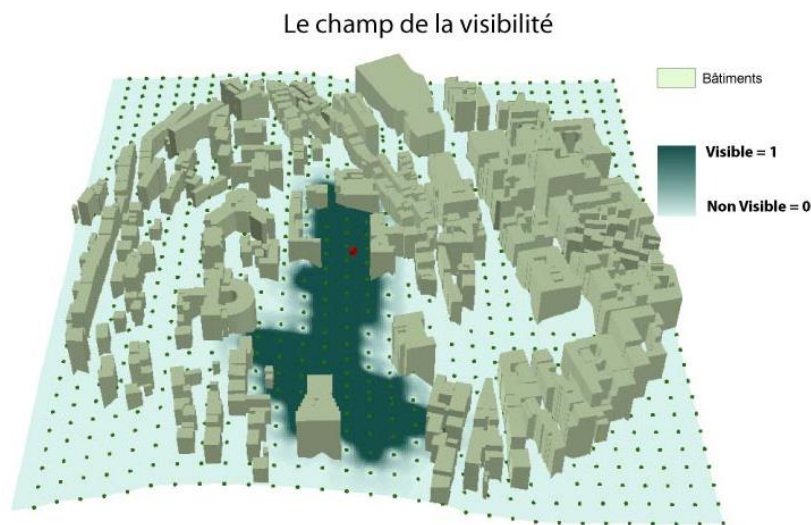


Figure 38. Le champ de la visibilité interpolé en utilisant la pondération inverse de distance (IDW), avec dans la table les bâtiments faisant obstacle à la vue et leur identification (une valeur de 0 signifie que c'est le sol qui fait écran)

### 3.4 L'effet d'un bâtiment sur l'intervisibilité

En employant cette méthode, nous avons également la possibilité de calculer l'effet sur l'intervisibilité d'un bâtiment spécifique. Le processus est simple : le bâtiment cible est choisi et l'effet de l'intervisibilité est calculé. Nous cherchons les points du maillage, qui n'ont pas d'intervisibilité à cause de ce bâtiment. Le résultat est un fichier 'shape' des points. Un nouvel attribut est créé dans la table qui indique si le bâtiment cible a un impact ou s'il n'a

pas d'impact. Si quelques points sont affectés, une autre liste d'attributs est ajoutée pour les points cachés à cause du bâtiment cible (Figure 39).

Attributs de VisGrîdeForBuilding							
FID	Shape	POINTID	GRID CODE	Z	Relative ZGH	Affected	HiddenPointsC
69	Point	8956	5472840	561,674	14,39	1	,8951, 8952, 8953, 9012, 9013,
77	Point	9013	5745781	574,6781	0,1	1	,8895, 8956,
100	Point	9134	5825875	582,5875	0,1	1	,8893, 8954,
89	Point	9074	5744918	574,5918	0,1	1	,8893,
111	Point	9192	5863129	586,4129	0,1	1	,8893,
122	Point	9249	5894113	589,5113	0,1	1	,8893,
80	Point	9016	5580632	558,1632	0,1	1	,8889,
0	Point	8636	5817961	581,8961	0,1	0	,
1	Point	8637	5832262	583,3262	0,1	0	,
2	Point	8638	5828164	582,9164	0,1	0	,

Enregistrement: 0 Afficher: Tout Sélectionnés. Enregistrements (0 sur 132 Sélectionnés.)

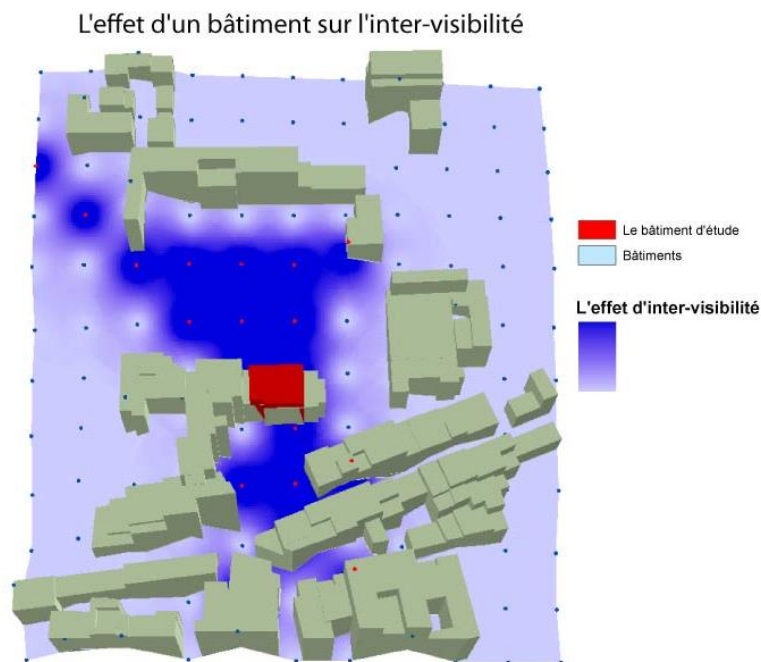


Figure 39. L'effet du bâtiment sur la carte interpolée par l'inter-visibilité en utilisant la pondération inverse de distance (IDW). Le champ de visibilité faible marque l'influence du bâtiment rouge, avec les points cachés en rouge aussi

Une autre application permet de calculer la visibilité du bâtiment pour les piétons passant dans la rue. Pour cela, nous ajoutons un maillage de points sur les façades du bâtiment d'intérêt. Ces points sont considérés comme des points vus, les points du maillage sur le terrain situé à un mètre et demi du sol étant considérés comme voyants, nous pouvons calculer ainsi le nombre des points visibles sur les façades de bâtiment pour chaque point du maillage de terrain (Figure 40).

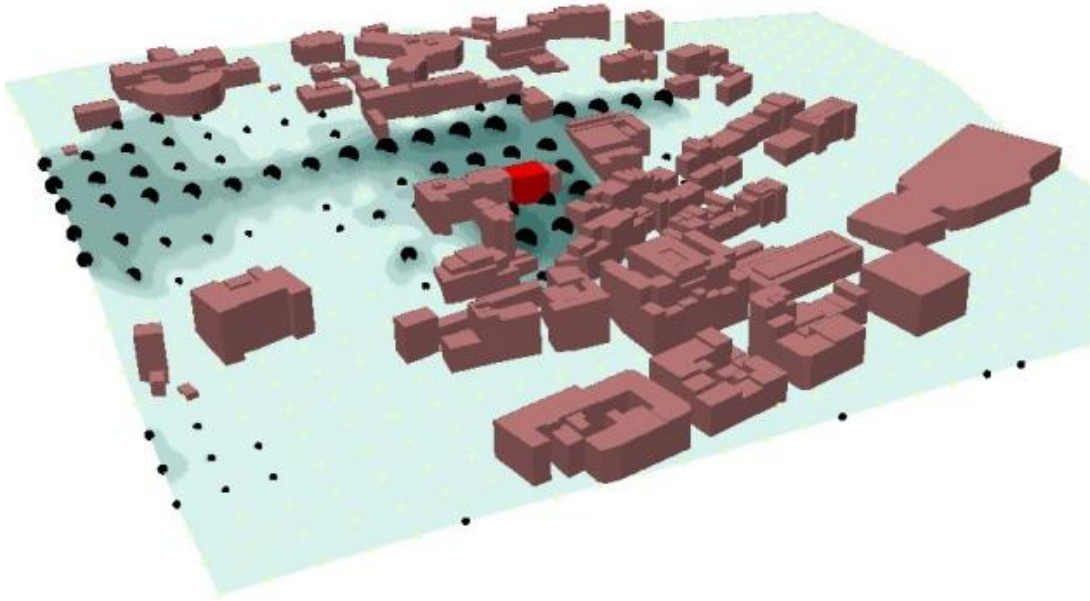


Figure 40. Carte de visibilité du bâtiment rouge. Les sphères sont proportionnelles à la visibilité du bâtiment.

### 3.5 Evaluation du résultat

Dans notre travail nous avons employé Matlab 7 comme un outil de simulation avec l'extension de l'outil de cartographie pour charger le fichier MNT, et les fichiers de forme des bâtiments, créer l'environnement 3D, calculer l'intervisibilité, calculer le champ de la visibilité, calculer l'effet d'un bâtiment sur l'intervisibilité et calculer la visibilité d'un bâtiment. Nous avons exporté le résultat sous la forme d'un fichier 'shape' de points avec les attributs qui stockent le pourcentage de la visibilité, les points visibles, les points cachés et les obstacles. Ce fichier peut être ouvert dans des outils classiques de SIG.

Cette méthode nous donne une nouvelle occasion de visualiser l'effet de la couche de bâtiment sur l'intervisibilité et le champ de la visibilité.

Le processus de calcul dépend de deux éléments :

- le nombre des polygones  $P$  dans l'environnement 3D (façades ou toits d'un bâtiment ou un triangle de TIN).
- le nombre des points du maillage de l'intervisibilité  $N$ .

La complexité du calcul du champ de visibilité est de l'ordre de  $N \cdot P$  intersections Segment-Polygone en 3D. La complexité du calcul de la carte d'intervisibilité est de l'ordre de  $N \cdot (N \cdot P)$  opérations d'intersection segment-Polygone en 3D. Il est donc très important de réduire le nombre de points  $N$  dans le maillage avant de construire la carte d'intervisibilité. Comme on l'a vu plus haut, le nombre de polygones dans l'environnement 3D joue un rôle essentiel dans la complexité du calcul de la visibilité. Ainsi devons-nous réduire le nombre de polygones autant qu'il est possible en conservant les formes principales de l'environnement 3D. Cette opération n'est pas obligatoire pour réaliser le calcul mais elle accélère de manière significative le processus.

### Simplification du MNT

Il y a plusieurs méthodes dans la littérature pour simplifier la compression de triangles de maille : Deering (1995), Touma and Gotsman (1998). Le MNT dans notre essai a été simplifié en groupant les points qui ont le même niveau de hauteur dans une marge d'une mètre puis en éliminant les points supplémentaires. L'idée ici est qu'il n'y a aucun besoin d'avoir 100 points de MNT pour décrire une surface plane. La taille d'intervalle peut être adaptée en fonction des besoins de l'utilisateur (Figure 41).

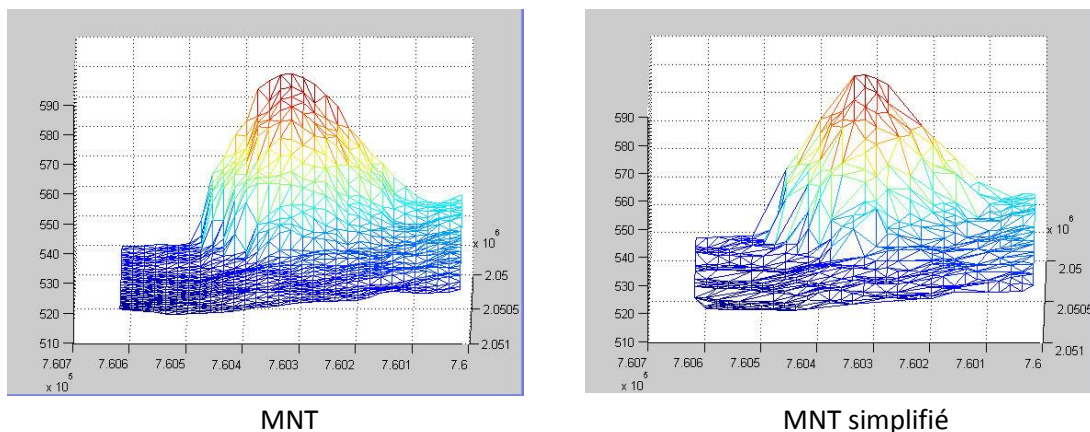


Figure 41. La simplification du MNT

### Simplification des polygones de bâtiment

Dans notre essai nous avons modélisé les bâtiments en polygones en utilisant Matlab. L'expérience montre que pour les segments qui composent les bâtiments issus d'une couche SIG, il y a souvent de nombreux sommets entre deux nœuds. Cela est dû au processus de numérisation. La simplification des polygones est traitée dans la littérature comme une

question de généralisation de la forme comme dans les travaux de Favier (1994), Anne Ruas (1999), Qingsheng, Brandenberger et Hurni (2002). Afin de simplifier les polygones des bâtiments, nous avons éliminé les points supplémentaires situés sur une même droite, et les points qui forment un arc de façade inférieur à 2 mètres. L'utilisateur peut modifier cette valeur (Figure 42).

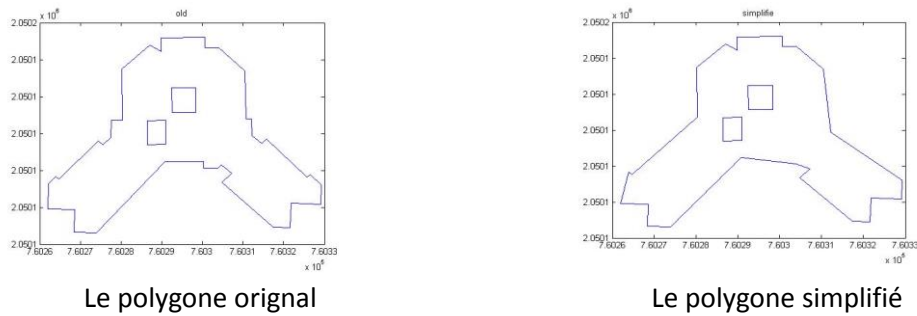


Figure 42. Exemple de simplification d'un polygone de bâtiment

Dans l'essai, une extraction d'une carte a été prise avec 88 points de maillage d'intervisibilité et 564 polygones 3D (424 façades de construction pour 63 bâtiments de polygone, 140 triangles pour le TIN). La méthode a été appliquée en utilisant Matlab 7 sur un PC avec un double noyau 4800+ d'AMD Athlon 30. Le temps de calcul était de 7 minutes. Avec les mêmes fichiers sans simplification le temps de calcul était de 16 mn. La précision visuelle du résultat final est impactée à un niveau de 1%.

### 3.6 Conclusion

Cette méthode est similaire à la modèle voxel présentée dans le chapitre 1, sauf qu'elle est fondée sur des données vectorielles généralisées de manière automatique par notre programme. Le résultat de cette méthode dépend du nombre de points dans notre maillage. Cette limitation nous a conduit à réfléchir à une autre méthode plus précise qui sera présentée dans le chapitre suivant.

# **Chapitre 4 Une nouvelle méthode pour calculer le champ de vision avec des données vectorielles de dimension 2 et 3**

Dans ce chapitre nous décrivons un nouvel algorithme d'isovist fondé sur des données SIG vectorielles en dimension 2 puis en dimension 3 et nous explorons les potentialités de cet algorithme pour la syntaxe spatiale présentée dans le chapitre 1. Le contenu de cette partie de la thèse a été publié comme un chapitre intitulé *A New Algorithm for 3D Isovists* dans l'ouvrage coordonné par Sabine Timpf et Patrick Laube : *Advances in Spatial Data Handling, Geospatial Dynamics, Geosimulation and Exploratory Visualization* 2013, pp 157-173.

## **4.1 Nouvel algorithme pour calculer le champ de visibilité en dimension 2**

Nous proposons un algorithme en dimension 2 fondé sur l'hypothèse qu'un espace en deux dimensions peut être envisagé comme un ensemble de segments sans intersection hormis leurs connexions aux extrémités. Les polygones du modèle de données originel sont alors décomposés en segments 2D. L'objectif est de rechercher parmi ces segments ceux qui bloquent la vue (Figure 43).

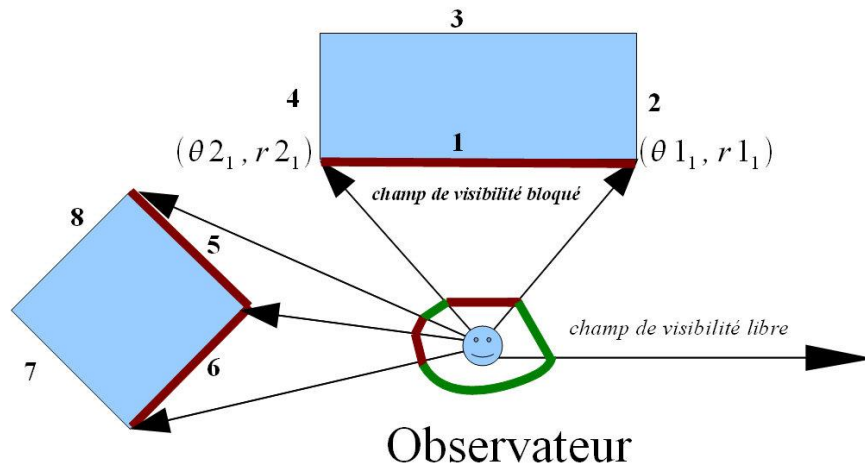


Figure 43. Les angles de vision en rouge sont associés aux segments qui limitent la vision dans ce sens, les angles en vert sont libres.

Si nous considérons cette liste des segments :

$$S = [\text{Segment}_1, \text{Segment}_2, \dots, \text{Segment}_N ]$$

Pour chaque segment ( $\text{Segment}_i$ ) nous avons deux extrémités  $[(\theta_{1_i}, r_{1_i}); (\theta_{2_i}, r_{2_i})]$  en système de coordonnées polaires centré sur le point de l'observateur.

*Définition 1* : un angle de vision de  $\text{Segment}_i$  est l'intervalle entre les angles de ses extrémités,  $AVS_i = [\theta_{1_i}, \theta_{2_i}]$ . Le module de cet angle de vision est :  $|AVS_i| = |\theta_{1_i} - \theta_{2_i}|$ .

*Définition 2* : un segment libre est un segment qui a deux extrémités visibles pour l'observateur (segments 1, 5, 6 de la Figure 43).

*Définition 3* : un champ de visibilité libre est associé à un angle de vision dans lequel aucun obstacle ne bloque la vue (les angles verts sur la Figure 43).

*Définition 4* : un champ de visibilité bloqué est associé à un angle de vision dans lequel il existe un obstacle qui bloque la vue (les angles rouges sur la Figure 43).

*Abréviations dans le pseudocode :*

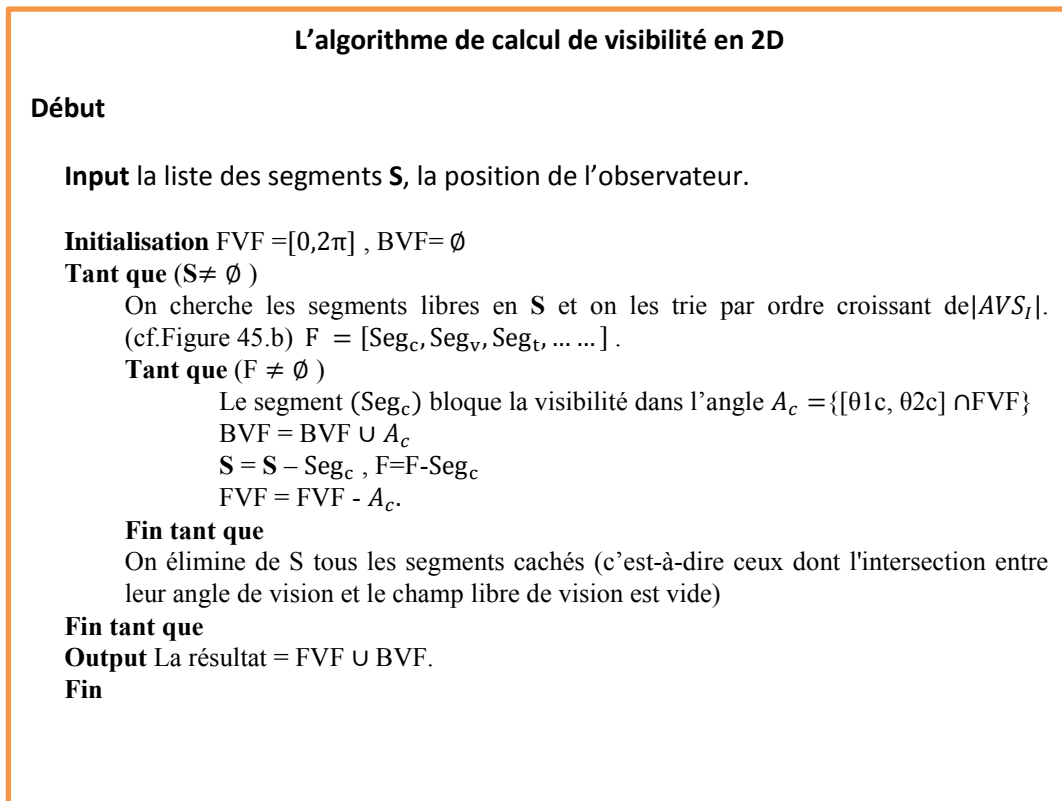
*FVF* : champ de visibilité libre.

*BVF* : champ de visibilité bloqué.

*S* : liste des segments.

*F* : liste libre des segments.

$|AVS_i|$  : valeur de l'angle de vision du segment  $\text{Segment}_i$ .



Le résultat de cet algorithme est une liste des intervalles angulaires. Chaque intervalle angulaire est associé soit au segment qui bloque la vision dans cette direction soit à l'espace libre. Cela signifie que l'isovist peut être considéré comme une agrégation d'éléments de la forme ci-dessous  $[0, 2\pi] * \{\text{Segments}\}$  (Figure 44).

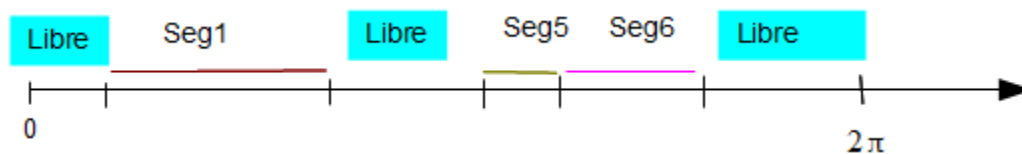


Figure 44. ISOVIST en environnement 2D.

L'algorithme trouve tous les segments qui limitent l'isovist. Le point critique de cet algorithme est de déterminer les segments qui bloquent la vision pour chaque extrémité de la liste. Supposons que nous ayons  $N$  segments qui forment les polygones, donc nous avons  $N$  extrémités. La complexité de cet algorithme sera d'ordre  $N^2$  opérations d'intersection entre deux segments 2D (avec  $N =$  nombre de segments). Ce calcul peut être amélioré par l'utilisation de l'algorithme « Sweep line algorithm » évoqué plus haut pour trouver l'intersection mutuelle entre les  $N$  segments. La complexité de cet algorithme pour  $N$  segments est de l'ordre de  $N \cdot \log(N)$ . En reliant les segments (ou les parties de segments) qui



bloquent le champ de visibilité, nous obtenons le polygone de visibilité défini par Rana et Batty (2004). La Figure 45 illustre un exemple de l'algorithme dans un environnement 2D.

Le résultat de la Figure 45 est:

$[0, \theta_{1_1}]$  est un espace libre.

$[\theta_{1_1}, \theta_{1_3}[$  est bloqué par (Segment<sub>1</sub>).

$[\theta_{1_3}, \theta_{2_3}]$  est bloqué par (Segment<sub>3</sub>).

$] \theta_{1_2}, \theta_{2_2}]$  est bloqué par (Segment<sub>2</sub>).

$] \theta_{2_2}, \theta_{2_1}]$  est bloqué par (Segment<sub>1</sub>).

$] \theta_{2_1}, \theta_{2_4}]$  est bloqué par (Segment<sub>4</sub>).

$] \theta_{2_4}, \theta_{2_5}]$  est bloqué par (Segment<sub>5</sub>).

$] \theta_{2_5}, 2\pi]$  est un espace libre.

La Figure 46 présente le résultat de notre méthode dans un environnement virtuel généré de manière aléatoire de dimension 2 en utilisant Matlab7.

La méthode employée ici ressemble du point de vue du résultat à celle de Do (1997). A la différence de celle-ci, elle a l'avantage de pouvoir être étendue à un environnement 3D. La Figure 47 présente les résultats de notre calcul appliqué à une base de données SIG 2D avec extraction du polygone de visibilité et production des quelques mesures morphométriques courantes de l'isovist listées par Rana (2006) que la méthode permet de produire. Dans la Figure 47, nous calculons le secteur visible, la distance, la direction au point le plus proche, la distance et la direction au point le plus lointain, l'index de circularité et l'angle de vue libre ou ouvert.

Analyse de visibilité et géolocalisation en milieu urbain avec un modèle numérique 3D

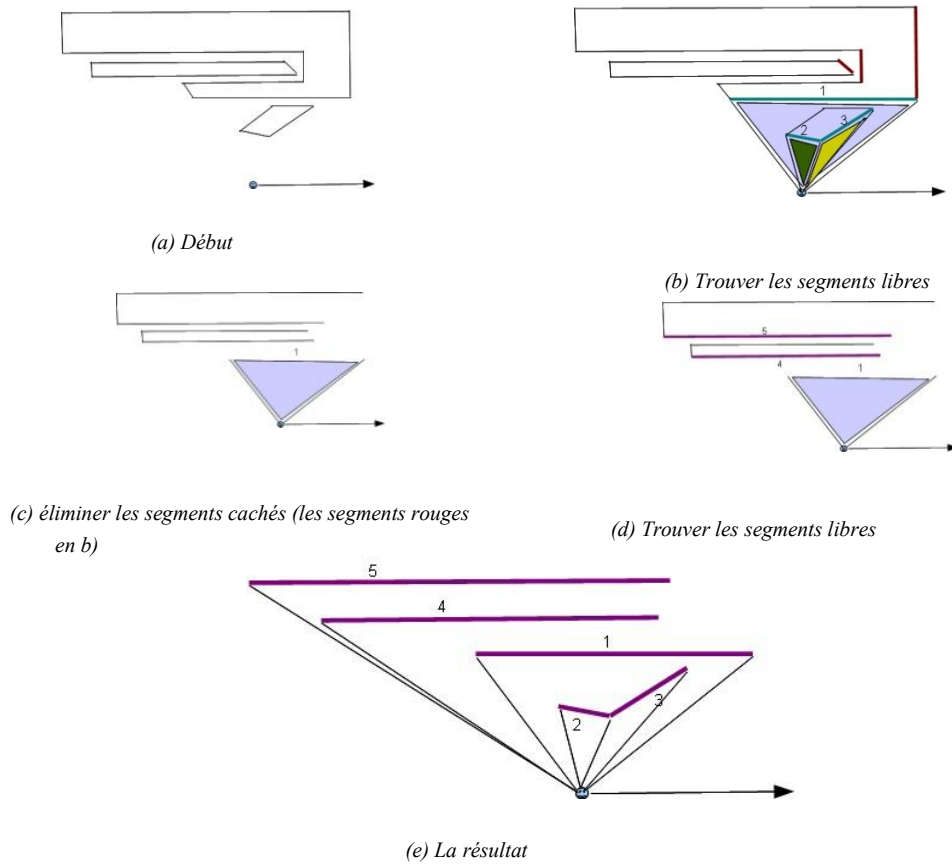


Figure 45. Les étapes de calcul de l'isovist en environnement 2D.

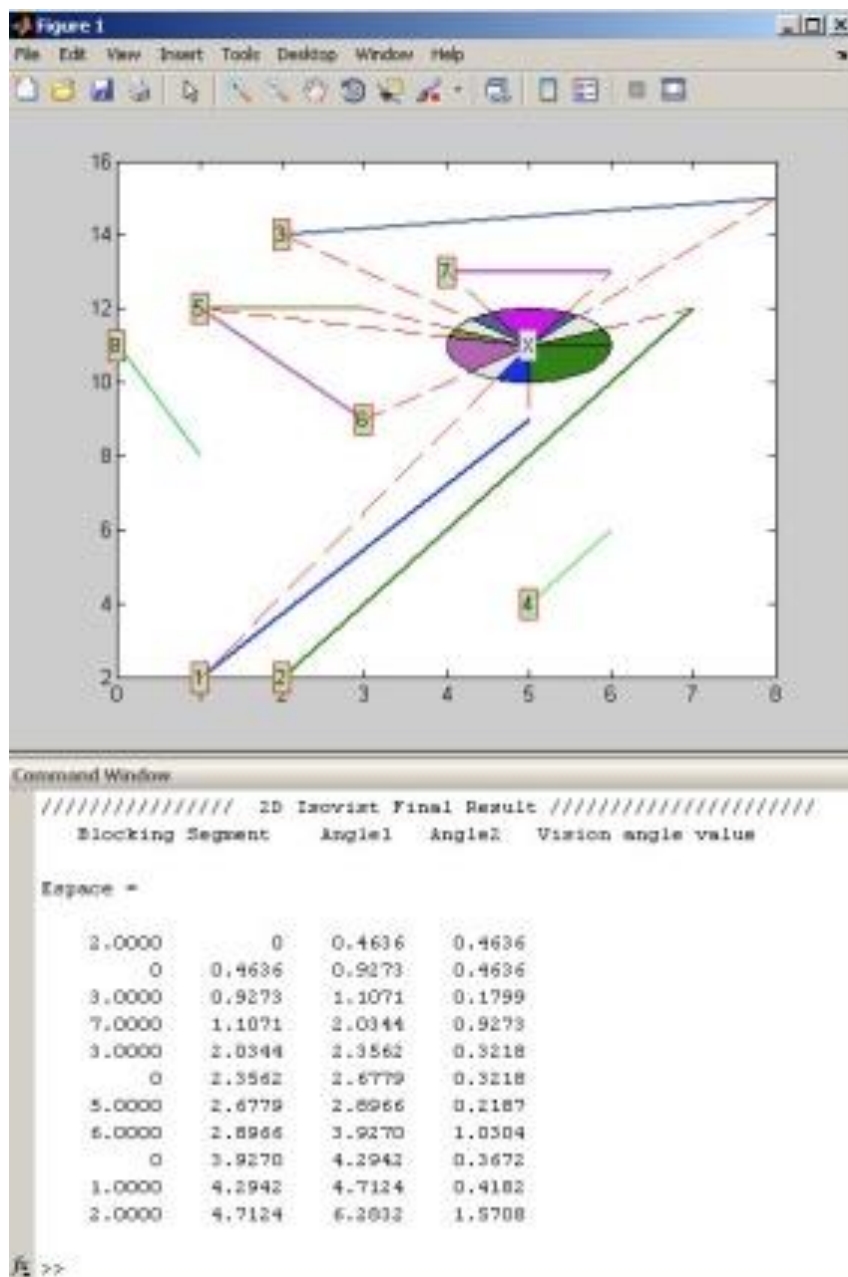


Figure 46. Le résultat de l'isovist en dimension 2 par un programme qui génère un environnement aléatoire 2D: chaque ligne contient l'identifiant du segment, le premier angle et le deuxième angle de l'intervalle de visibilité et la valeur de l'angle de vision.

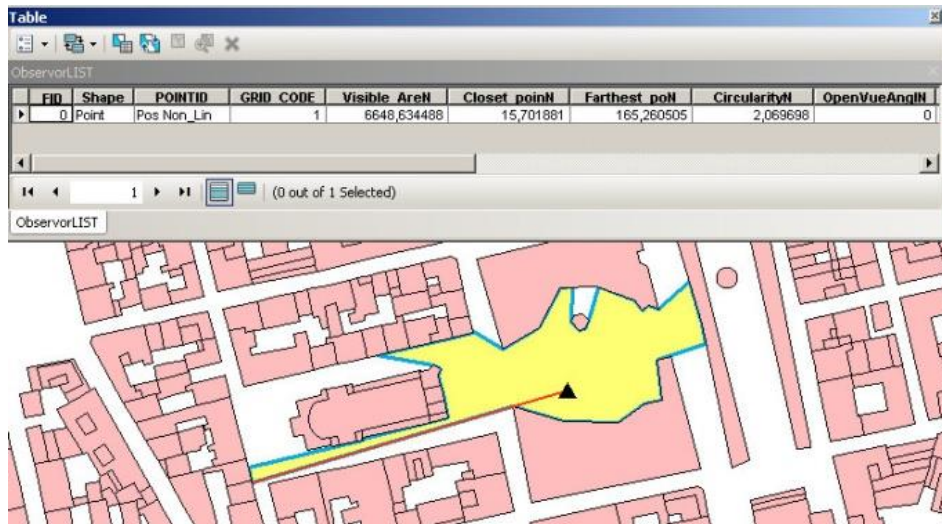


Figure 47. L'isovist en dimension 2 exporté en fichier shape dans ArcGIS. Le triangle représente l'observateur, la surface jaune l'espace visible, le contour bleu le polygone de visibilité, le ligne rouge présente la distance au point visible le plus lointain.

Cet algorithme nous donne la possibilité de calculer un grand nombre de paramètres à propos des objets visibles par l'observateur. La Figure 48 présente le calcul du champ de vision dans un environnement en dimension 2 composé de bâtiments du centre-ville de Saint-Etienne avec leurs attributs associés décrivant les types de commerce qui s'y trouvent. Le calcul fait apparaître que les magasins les plus visibles par l'observateur sont du type « Equipement de la personne » avec un pourcentage de 61%. La visibilité libre est de 5%. Pour l'angle de vue de détail présenté dans la Figure 48, les magasins visibles qui dominent sont de type «Alimentaire » (17% du total), et le type dominant des bâtiments visibles est « Résidence » avec 54%.

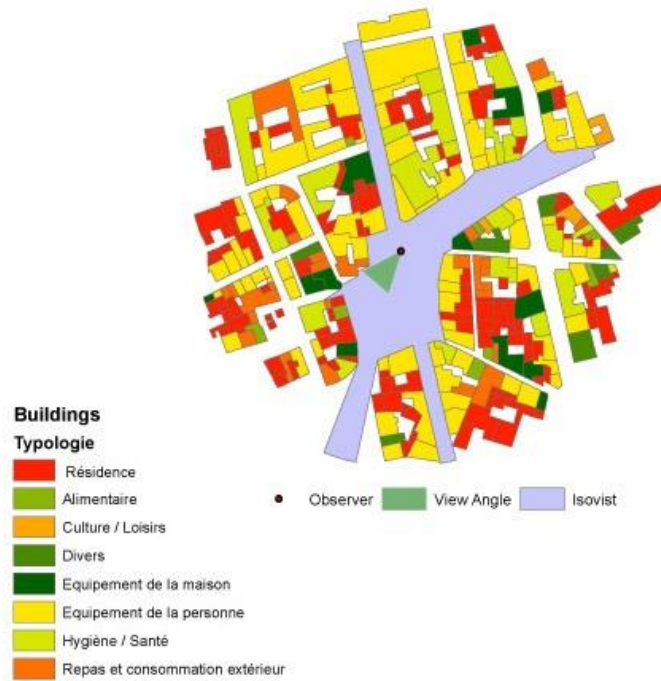


Figure 48. Le champ de visibilité en dimension 2 avec le type des bâtiments et l'angle de vue

Pour faciliter l'utilisation de notre méthode, nous avons développé une application en VB.NET utilisant la bibliothèque gratuite ShapeDLL<sup>21</sup>. Ce programme n'a pas besoin des bibliothèques de ArcGIS ou de Matlab. Le programme permet de télécharger un fichier shape, de l'afficher et d'identifier la position de l'observateur à la souris ou à partir d'un fichier shape (Figure 49).

<sup>21</sup> <http://arcscripts.esri.com/details.asp?dbid=11810>

## Analyse de visibilité et géolocalisation en milieu urbain avec un modèle numérique 3D

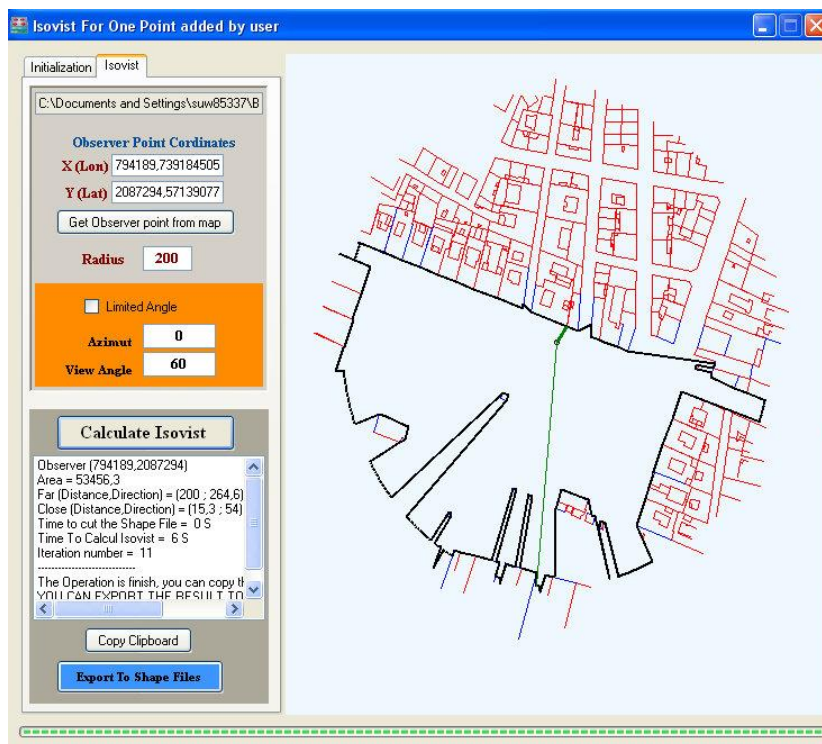
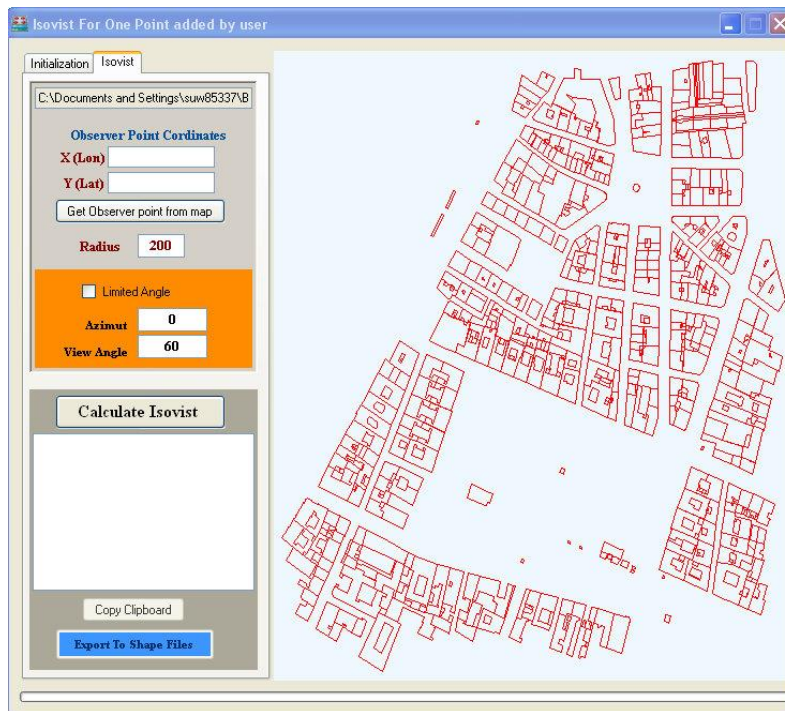


Figure 49. Le champ de visibilité en dimension 2. Vue de notre programme en VB.NET

Le programme permet aussi de calculer la visibilité suivant une direction et un angle de vision donnés pour simuler par exemple une prise de vue photographique (Figure 50). Le programme calcule aussi différentes statistiques citées plus haut comme la surface du champ de visibilité, le point le plus proche avec la direction dans laquelle il se trouve, le point le plus loin avec aussi sa direction. Le résultat peut être exporté en format 'shape' pour le visualiser dans ArcGIS ou dans un autre programme SIG.

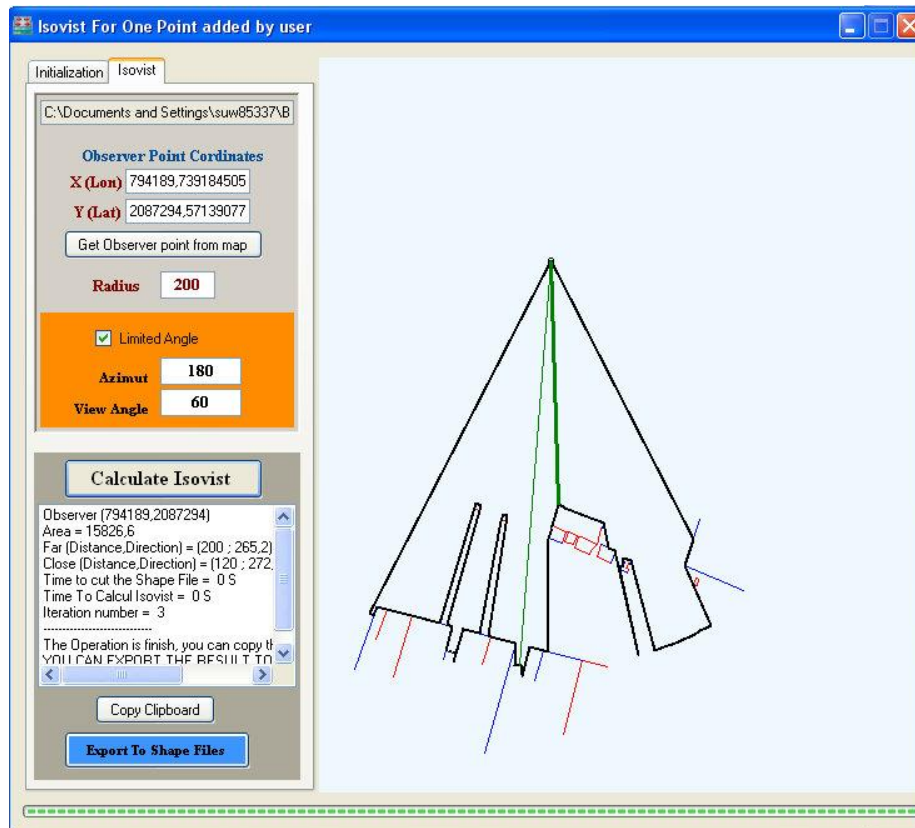


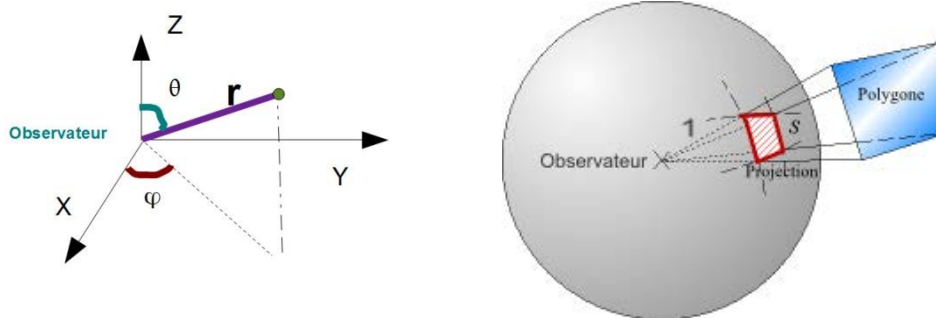
Figure 50. Le champ de vision avec une direction et angle de vue.

Le programme peut aussi lire une série de points d'observateur à partir d'un fichier shape avec une direction de vision et un champ de vision pour chaque point, calculer les champs de vision correspondant et exporter le résultat en shape file.

## 4.2 Un nouvel algorithme pour calculer le champ de visibilité en dimension 3

En généralisant notre algorithme précédent en 3 dimensions, nous considérons l'environnement 3D comme une agrégation de polyèdres qui sont décomposés en polygones 3D. La Figure 51.a présente un système de coordonnées sphériques centré sur le point de l'observateur. Chaque point est défini par  $(r, \varphi, \theta)$  avec  $r \in \mathbb{R}$ ,  $\varphi \in [0, 2\pi]$ ,  $\theta \in [0, \pi]$ . Le polygone  $\text{Polygone}_i$  est défini par une liste ordonnée des points  $\{p1_i, p2_i, \dots, pn_i\}$ , qui est définie par des triplets :  $\{(r1_i, \varphi1_i, \theta1_i), (r2_i, \varphi2_i, \theta2_i), \dots, (rn_i, \varphin_i, \thetan_i)\}$ . La Figure 51.b présente la projection d'un polygone sur la sphère unitaire (de rayon égale à 1). La projection du polygone  $\text{Polygone}_i$  est composée des points  $\{(1, \varphi1_i, \theta1_i), (1, \varphi2_i, \theta2_i), \dots, (1, \varphin_i, \thetan_i)\}$ , avec les arcs sphériques qui connectent la liste des points dans l'ordre :

$\{(1, \varphi1_i, \theta1_i), (1, \varphi2_i, \theta2_i)\}; \dots; \{(1, \varphi(n-1)_i, \theta(n-1)_i), (1, \varphin_i, \thetan_i)\}$ .



(a) Les coordonnées sphériques

(b) Source Wikipédia modifié par Suleiman (les arcs sphériques sont en rouge)

Figure 51. La projection d'un polygone en coordonnées sphériques

Puisque le rayon est égal à 1 pour tous les points, nous pouvons définir la projection comme une série de couples :  $\{(\varphi1_i, \theta1_i), (\varphi2_i, \theta2_i), \dots, (\varphin_i, \thetan_i)\}$ . La projection forme l'angle solide de visibilité du  $\text{Polygone}_i$  et la surface de la projection est la valeur de l'angle solide de visibilité du  $\text{Polygone}_i$ .

*Définition 5:  $SAVP_i$  est l'angle solide de visibilité pour le  $\text{Polygone}_i$ .  $|SAVP_i|$  est la valeur de l'angle de visibilité.*



*Définition 6: un polygone libre est un polygone dont les côtés (les segments de sa frontière fermée) sont visibles par l'observateur.*

*Définition 7: un isovist libre est associé à un angle solide pour lequel aucun obstacle ne peut gêner la visibilité.*

*Définition 8: un isovist bloqué est associé à un angle solide pour lequel un obstacle empêche la visibilité.*

Abréviations dans le pseudocode :

*FS : surface associée au champ de visibilité libre sur la sphère unité.*

*BS : surface associée au champ de visibilité bloqué sur la sphère unité.*

*P : liste des polygones.*

*F : liste des polygones libres.*

*|SAVP<sub>l</sub>| : module de l'angle solide de la visibilité pour la Polygone<sub>l</sub>.*

### L'algorithme de visibilité en 3D

#### Début

**Input** la liste des polygones **P**, la position de l'observateur.

**Initialisation** FS= {Le sphère d'unité entière}, BS =  $\emptyset$  .

**Tant que** (**P**  $\neq \emptyset$ )

Trouver les polygones libres en **P** et on les trie par ordre croissant de |SAVP<sub>l</sub>|, F = [Polygone<sub>c</sub>, Polygone<sub>v</sub>, Polygone<sub>t</sub>, ... ..].

**Tant que** (F  $\neq \emptyset$ )

Le polygone (Polygone<sub>c</sub>) bloque la visibilité dans l'angle solide SA<sub>c</sub> = SAVP<sub>c</sub>  $\cap$  FS.

BS = BS  $\cup$  SA<sub>c</sub>

**P** = **P** - Polygone<sub>c</sub> , F = F - Polygone<sub>c</sub>.

FS=FS - SA<sub>c</sub> .

**Fin tant que**

Élimine tous les polygones cachés de **P**. (c'est-à-dire ceux dont l'intersection entre leur angle de vision et le champ libre de vision est vide)

**Fin tant que**

**Output** La résultat = FS  $\cup$  BS.

**Fin**

Cet algorithme produit un ensemble des angles solides associés aux identificateurs des polygones de l'espace. L'algorithme procède à des tests d'intervisibilité entre la limite de chaque polyèdre et le point de l'observateur en choisissant un nombre constant  $t$  de points sur cette limite. La complexité de l'algorithme est donc d'ordre  $P$  opérations d'intersection entre

un polygone et un segment en environnement 3D. La Figure 52.a présente le résultat de notre algorithme de calcul de visibilité sur un cube 3D virtuel en utilisant l'environnement de développement de Matlab.

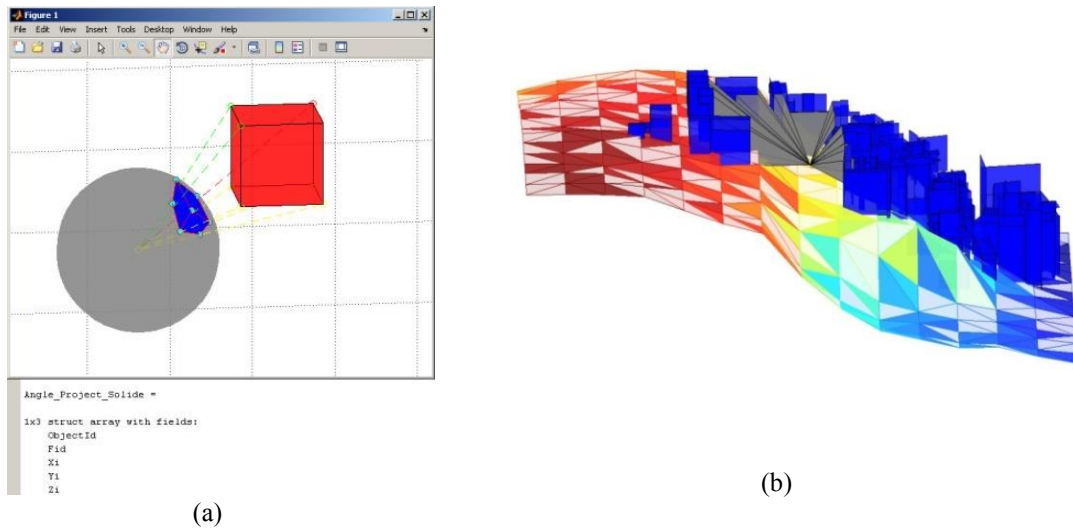
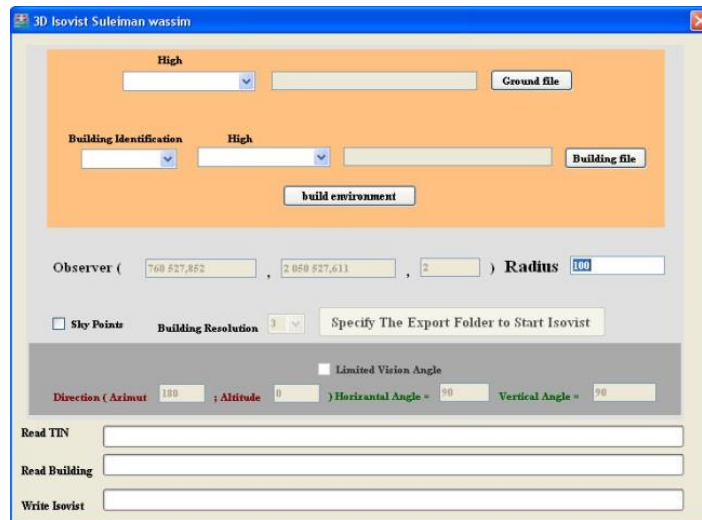


Figure 52. La visibilité 3D avec des données SIG 3D en utilisant Matlab, (a) Visibilité d'un cube virtuel, (b) Environnement 3D des bâtiments sur relief. La couleur grise présente la visibilité.

## 4.3 L'application dans un environnement 3D

Des tests ont été réalisés avec les bâtiments 3D situés sur une zone de collines à Saint-Etienne (France). Le test utilise les données SIG 3D des bâtiments issues de la base de données de l'IGN et un Modèle Numérique de Terrain de résolution un mètre pour le terrain, issu d'une campagne de mesure aéroportée. Nous utilisons la triangulation de Delaunay pour modéliser le terrain comme un ensemble de triangles. Les bâtiments sont transformés en un ensemble de rectangles représentant le toit et les façades (Suleiman, Joliveau et Favier 2011). Notre algorithme calcule la visibilité 3D à partir de n'importe quel point de vue situé sur le sol à une hauteur donnée. Le code a été écrit en utilisant les fonctions spatiales et géométriques de Matlab 7. Le rayon de la zone d'étude est de 500 mètres et le modèle contient 17 bâtiments (231 façades). Le temps d'exécution est de 1,5 minutes sur un AMD Athlon 30 Dual Core 4800+. Les résultats sont illustrés comme un ensemble d'angles solides avec leurs polygones correspondants (façade de bâtiment ou triangle de terrain) qui bloquent la visibilité dans cette

direction (Figure 52.b). La Figure 53 présente les résultats de notre programme exportés dans ArcGIS et visualisés avec Arcscene.



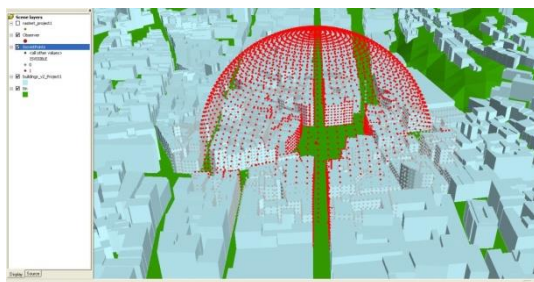
L'interface de calculer l'isovist en 3D



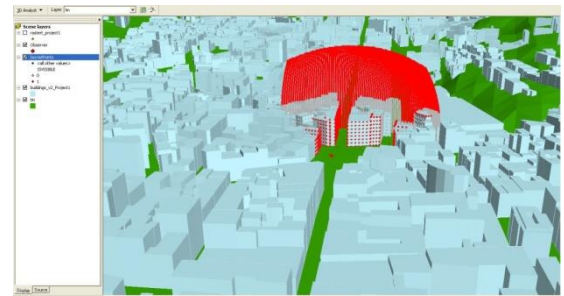
(a) Test avec relief



(b) Test avec sol plat



(c) Test avec sol plat et point dans l'espace



(d) Test avec angle de vue

Figure 53. Les résultats du calcul de visibilité en 3D exportés en fichiers shape et visualisés dans ArcGIS. Les points correspondant aux angles solides de visibilité sont ajoutés automatiquement selon un treillis régulier sur le sol, dans les airs et sur les façades des bâtiments. Les points rouges sont les points visibles par l'observateur.

Pour visualiser le cône de la visibilité, on a utilisé le format de données 3D ArcGIS multipatch<sup>22</sup>. On a développé un Plugin Arcscene en C# qui nous donne le cône de visibilité à

<sup>22</sup> <http://www.esri.com/library/whitepapers/pdfs/multipatch-geometry-type.pdf>

partir des points de visibilité calculés en Figure 53. La Figure 54 présente le cône de visibilité en format mutipatch.

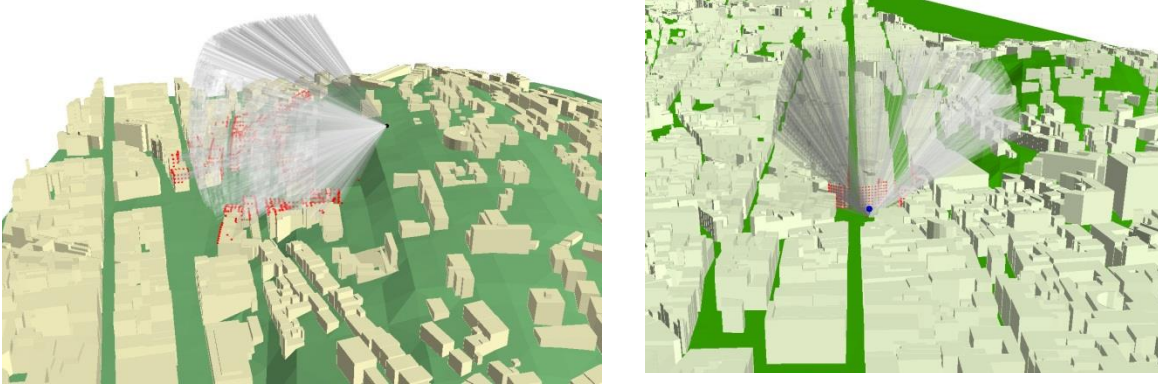


Figure 54 le cône de visibilité présenté par le couleur gris transparente en format ArcGIS multipatch. Les points rouges sont les points visibles sur les bâtiments.

#### **4.3.1 La combinaison avec les photographies**

Pour visualiser notre résultat de manière plus réaliste, on peut utiliser la matrice de transformation d'une caméra, calibrée afin d'obtenir l'image de synthèse de notre calcul. Puis on peut comparer notre image synthèse avec une photographie prise de la même position et dans la même direction que l'image de synthèse. La Figure 55 illustre la similarité entre notre image de synthèse et la photographie.

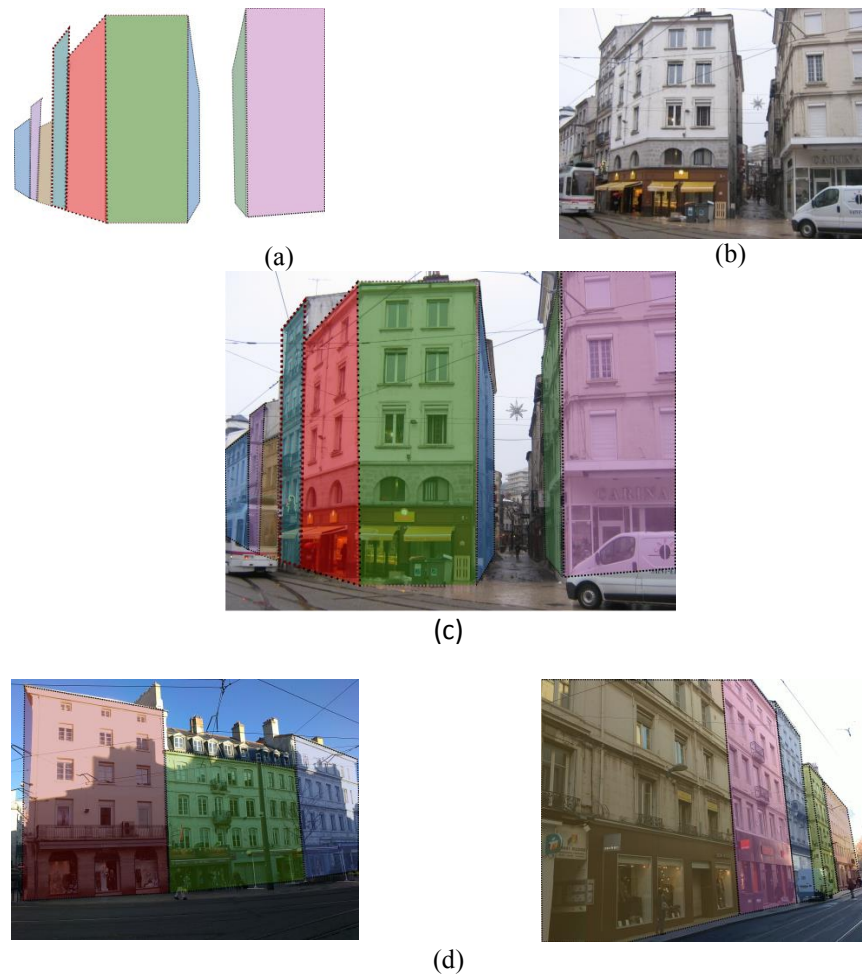


Figure 55 (a) L'image synthèse de l'isovist, (b) L'image photographique, (c) L'appariement entre l'image synthèse et l'image réelle (d) Autre exemples

Les méthodes d'appariement sont basés sur l'appariement des primitives (coins, lignes) entre l'image synthèse et l'image photographique.

### 4.3.2 La syntaxe spatiale

Notre méthode fondée sur une description de l'isovist par l'angle solide nous permet de calculer des indices pour la syntaxe spatiale différents de ceux de Bilsen (2010). La Figure 56 illustre l'espace ouvert ou la vue vers le ciel calculé sur 100 mètres. L'espace ouvert est pondéré par l'angle solide libre.

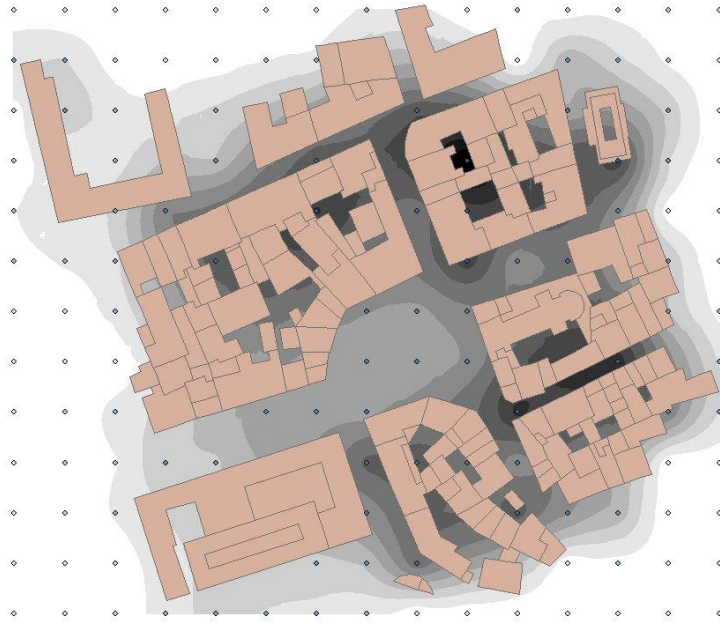
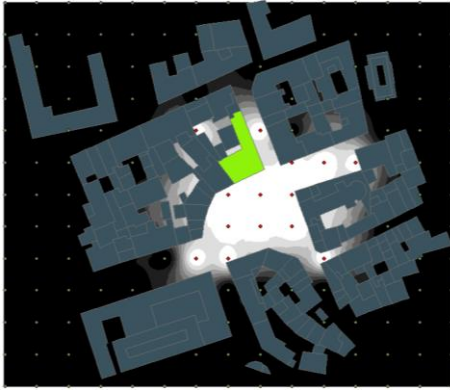
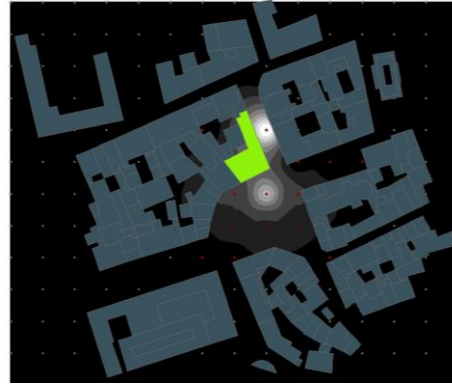


Figure 56. L'espace ouvert calculé à 100 mètres, la partie claire est la partie où l'espace ouvert est élevé.

Ce calcul appelé *Space Open Index* est calculé dans Fisher-Gewirtzman et al (2005) par lancement des rayons. Ce calcul nous permet de calculer la visibilité d'un bâtiment identifié à partir du sol, puis de faire l'interpolation par la pondération de l'inverse de la distance (IDW). La Figure 57.a est calculée avec un poids de 1 pour les points voyant du bâtiment, 0 pour les points non-voyants. Dans ce cas la zone claire marque l'espace où l'on peut voir le bâtiment vert. La Figure 57.b est calculée avec un poids égal à l'angle solide de la visibilité du bâtiment. L'indice mesure l'intensité de la présence du bâtiment dans l'espace visible à partir d'un point donné. Si on se trouve dans la partie claire, cela signifie que le bâtiment vert occupe une grande partie du champ de visibilité, et inversement que le bâtiment vert est peu ou pas visible dans la partie sombre.



(a) Pondération point visible égale à 1 point no visible égale à 0



(b) Pondération suivant l'angle solide de la visibilité

Figure 57. La visibilité du bâtiment vert. Les points rouges sont les points visibles et la partie claire correspond à l'interpolation de l'espace de la visibilité par distance inverse pondérée (IDW)

## 4.4 Conclusion

Dans ce chapitre nous avons présenté une nouvelle méthode pour calculer l'isovist avec des données vectorielles dans un environnement 2D et 3D qui prend en compte les formes du relief et les formes géométriques des bâtiments. Nous avons proposé un nouvel algorithme 2D et 3D et l'avons implémenté avec Matlab et .Net. Cette méthode nous donne la possibilité de calculer de nouveaux indices utiles en syntaxe spatiale, tel que le pourcentage des bâtiments historiques ou l'espace qu'on peut voir à partir d'un point donné.

Les premiers tests démontrent la faisabilité et l'intérêt de cette approche en 2D comme en 3D. Une évaluation réelle de cette proposition en termes de temps de calcul nécessiterait un développement spécifique optimisant le matériel (la carte graphique) ou réalisée dans un langage de programmation de bas niveau. Les tests ont été réalisés sur des modèles simples de bâtiment de type LoD1 en City GML, à toit plat, sans décrochement ni chien-assis. La précision des isovists calculés reflète ces limites mais la méthode proposée peut accepter en entrée n'importe quelle forme de bâtiment dès lors qu'il est possible de la décomposer en polygones. La transparence est une limite pour notre algorithme. Tous les objets sont considérés comme solides et opaques. Une amélioration en ce sens est nécessaire pour prendre en compte par exemple les surfaces vitrées non réfléchissantes et surtout la végétation.

# Conclusion générale

Nous avons présenté dans cette thèse une méthode pour calculer la visibilité à partir d'un point donné dans un environnement 3D, et une méthode pour géolocaliser la prise de vue en utilisant la signature des bâtiments. Dans le cadre de cette recherche en géomatique portant sur la visibilité et la localisation en milieu urbain, il reste encore de nombreux problèmes à régler.

Bien sûr, les réponses que nous apportons, fondées sur l'usage de données liées aux bâtiments, qu'il s'agisse de la visibilité ou de la localisation de photographies, concernent essentiellement les milieux urbains. Les espaces ruraux ou naturels posent des problèmes différents qui nécessitent des recherches dans d'autres directions. Même dans les milieux urbains, la présence d'arbres devant et entre les bâtiments est un problème. Ils peuvent cacher les bâtiments dans les images et donc rendre leur reconnaissance compliquée. De plus, la présence d'arbres dans la base des données 3D étant rare et souvent mal renseignée, le calcul de la visibilité peut manquer de précision. Mais aujourd'hui les données lidar sont de plus en plus disponibles, ce qui donnera dans le futur la disponibilité d'environnements 3D plus précis et intégrant une couche de végétation. La photogrammétrie est aussi une piste importante pour construire un SIG 3D fiable et bon marché.

Pour la géolocalisation par la texture des bâtiments en milieu urbain, on a fait l'hypothèse que nous disposions d'une base de données de signatures de texture produite manuellement. L'apparition de modèles 3D texturés détaillés associés aux globes virtuels (Google a Street View, Microsoft a Bing StreetSide), développés pour des sites particulier comme celui du groupe PagesJaunes avec la technologie UrbanDrive, ou même mis en œuvre de plus en plus souvent directement par les villes et les collectivités, offre des opportunités importantes pour une extraction systématique de textures. Pour ce qui concerne la visibilité, nos méthodes ont été testées en utilisant un modèle 3D simple (LOD1<sup>23</sup>). Or les modèles numériques de terrain et de bâtiments deviennent toujours plus précis. Il existe aussi des méthodes qui deviennent plus accessibles pour construire des modèles 3D, comme la stéréovision (Remblie 2011). Suivant Jia Yi, Saniie et Oruklu (2012), on peut aussi utiliser à cet effet une caméra bon marché comme la *Kinect*. Ainsi, les calculs de visibilité dans un environnement synthétique deviennent plus conformes à la réalité. Les simulations de zones vues pourront se réaliser

---

<sup>23</sup> [http://www.citygmlwiki.org/index.php/Basic\\_Information](http://www.citygmlwiki.org/index.php/Basic_Information)



dans des rendus plus réalistes, appuyés sur des bases de données 3D produites à l'aide d'un moteur 3D<sup>24</sup> (Figure 58).



Modèle 3D par sketchup<sup>25</sup>



Photo réel par Google Street View

Figure 58 modèle 3D texturé d'un bâtiment

Une autre piste de recherche concerne les données vidéo. La vidéo, définie comme une série d'images continue dans le temps et dans l'espace auxquelles s'ajoutent des données audio, est beaucoup plus riche qu'une seule image. Notre étude sur la localisation d'une image peut être une première étape dans une recherche future pour la localisation de vidéos.

L'analyse de visibilité réaliste est une question importante de recherche qui peut trouver des applications dans de nombreux domaines : l'architecture, l'urbanisme, le paysage, la sécurité, la conception de réseaux sans fil, la gestion du paysage, la mise en œuvre d'accès piétonniers, la perception humaine de l'espace ... La prise en compte de la troisième dimension dans le calcul de visibilité était un réel défi que nous avons tenté de relever. Les résultats de nos travaux sont prometteurs et les avancées technologiques permettent d'envisager dès maintenant des développements industriels.

<sup>24</sup> [http://fr.wikipedia.org/wiki/Moteur\\_3D](http://fr.wikipedia.org/wiki/Moteur_3D)

<sup>25</sup> <http://www.sketchup.com/>

# Références

## Références du chapitre 1

- Amlacher K., Paletta L., Luley P., Fritz G., Almer A., 2008, Geo-Indexed Object Recognition in Urban Scenarios, *32nd Workshop of the Austrian Association for Pattern Recognition*, 2008, p. 9–18.
- Attia Youssef, 2012. Interfaçage de bases de données photographiques et géographiques par appariement de lignes. Thèse de Doctorat. Saint-Etienne : Université de Saint-Etienne, 180 p.
- Ashihara Y., 1984, The Aesthetic Townscape, September 1984, p. 195–139.
- Bartie P., Mills S., Kingham S., 2008, An Egocentric Urban Viewshed: A Method for Landmark Visibility Mapping for Pedestrian Location Based Services, in *Geospatial Vision*, Springer Berlin Heidelberg, p. 61–85. < <http://www.springerlink.com/content/k421132277816363/abstract/> >
- Bartie P., Reitsma F., Kingham S., Mills S., 2010, Advancing visibility modelling algorithms for urban environments, *Computers, Environment and Urban Systems*, November 2010, vol. 34, n°6, p. 518–531. < <http://dx.doi.org/10.1016/j.compenvurbsys.2010.06.002> >
- Batty M., Jiang B., 1999, Multi-agent simulation: new approaches to exploring space-time dynamics in GIS, < <http://eprints.ucl.ac.uk/268/> >
- Batty M., 2001, Exploring isovist fields: space and shape in architectural and urban morphology, *Environment and Planning B: Planning and Design*, 2001, vol. 28, n°1, p. 123–150.
- Bay H., Ess A., Tuytelaars T., Van Gool L., 2008, Speeded-Up Robust Features (SURF), *Computer Vision and Image Understanding*, June 2008, vol. 110, n°3, p. 346–359. < <http://dx.doi.org/10.1016/j.cviu.2007.09.014> >
- Benedict M. L., 1979, To Take Hold of Space: Isovists and Isovists Fields. *Environment and Planning*, 1979, p. 6, 47–65.
- Bentley, Ottmann, 1979, Algorithms for Reporting and Counting Geometric Intersections, *IEEE Transactions on Computers*, September 1979, vol. C-28, n°9, p. 643–647. < <http://dx.doi.org/10.1109/TC.1979.1675432> >
- Bilsen V., 2009, How can serious games benefit from 3D visibility analysis?, in *International Simulation and Gaming Association Conference*, Singapore.
- Bilsen V., 2010, 3D Visibility Analysis in Virtual Learning Environments and Interactive and Digital Media, in *Interactive & Digital Media for Education in Virtual Learning Environment*, New York.
- Bioret N., Moreau G., Servières M., 2009, Géolocalisation en milieu urbain par appariement entre une collection d'images et un SIG 2D, *Ingénierie des systèmes d'information*, 2009, vol. 14, n°5, p. 107–131. < <http://dx.doi.org/10.3166/isi.14.5.107-131> >
- Brossard T., Joly D., Tourneux F., 2008, Modélisation opérationnelle du paysage, in *Paysage et information géographique*, Lavoisier, p. 117–137.
- Cavailhès J., Brossard T., Hilal M., Joly D., Tourneux P.-F., Tritz C., Wavresky P., 2007, Le prix des paysages

périurbains, *Économie rurale*, 6 April 2007, n°297-298, p. 71–84.

Christenson M., 2010, Registering visual permeability in architecture: isovists and occlusion maps in AutoLISP, *Environment and Planning B: Planning and Design*, 2010, vol. 37, n°6, p. 1128 – 1136.  
< <http://dx.doi.org/10.1068/b36076> >

Conroy Dalton R., Bafna S., 2003, The syntactical image of the city: a reciprocal definition of spatial elements and spatial syntaxes. < <http://eprints.ucl.ac.uk/1104/> >

Debevec P. E., Taylor C. J., Malik J., 1996, Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach, in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, p. 11–20. < <http://dx.doi.org/10.1145/237170.237191> >

Do E. Y.-L., 1997, Tools for visual and spatial analysis of CAD models, *CAAD Futures 1997 Conference*, 4 August 1997, p. 373–388.

Do E. Y.-L., 1994, Design and description of form -- using tool command language Tk/Tcl to visualize isovist by lighting and shadow casting analogy., *Computer program ARCH8193A4*, 1994.

Do E. Y.-L., 1994, Isovist calculation in AutoCAD., *Computer program and independent study*, 1994.

Do E. Y.-L., 1995, Visual Analysis through Isovist -- building a computation tool, *Working paper and computer program*, 1995.

Fischler M., Bolles R., 1981, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM*, June 1981, vol. 24, n°6, p. 381–395.

Fisher-Gewirtzman D., Shach Pinsky D., Wagner I. A., Burt M., 2005, View-oriented three-dimensional visual analysis models for the urban environment, *Urban Design International*, April 2005, vol. 10, p. 23–37.  
< <http://dx.doi.org/10.1057/palgrave.udi.9000133> >

Fishman J., Haverkort H., Toma L., 2009, Improved visibility computation on massive grid terrains, ACM Press, p. 121. < <http://dx.doi.org/10.1145/1653771.1653791> >

Floriani L. D., Magillo P., 2003, Algorithms for visibility computation on terrains: a survey, *Environment and Planning B: Planning and Design*, 2003, vol. 30, n°5, p. 709 – 728. < <http://dx.doi.org/10.1068/b12979> >

Floriani L. D., Magillo P., 1994, Visibility algorithms on triangulated digital terrain models, *International journal of geographical information systems*, January 1994, vol. 8, n°1, p. 13–41.  
< <http://dx.doi.org/10.1080/02693799408901985> >

Franklin W. R., Ray C. K., 1994, Higher isn't Necessarily Better: Visibility Algorithms and Experiments, *ADVANCES IN GIS RESEARCH: SIXTH INTERNATIONAL SYMPOSIUM ON SPATIAL DATA HANDLING*, 1994, vol. 5, p. 751–770.

Gibson J. J., 1983, *The Senses Considered as Perceptual Systems*, 1st ed. Greenwood Press Reprint,

Gueymard S., 2006, Facteurs environnementaux de proximité et choix résidentiels, *Développement durable et territoires*, 4 May 2006. < <http://developpementdurable.revues.org/index2716.html> >

Haralick, Shanmugam, Dinstein, 1973, Textural Features for Image Classification, *Systems, Man and Cybernetics, IEEE Transactions on*, 1973, vol. 3, n°6, p. 621, 610.

Hoiem D., Efros A. A., Hebert M., 2007, Recovering Surface Layout from an Image, *Int. J. Comput. Vision*, October 2007, vol. 75, n°1, p. 151–172. < <http://dx.doi.org/10.1007/s11263-006-0031-y> >

Hillier B., Hanson J., 1984, *The Social Logic of Space*, First ed. Cambridge University Press.

Ittelson W., 1961, Visual Space Perception. William H. Ittelson. Springer, New York, 1960. 212 pp. \$6, *Science*, 21 April 1961, vol. 133, n°3460, p. 1241–1242. < <http://dx.doi.org/10.1126/science.133.3460.1241-c> >

Jim C. Y., Chen W. Y., 2006, Impacts of urban environmental elements on residential housing prices in Guangzhou (China), *Landscape and Urban Planning*, 28 November 2006, vol. 78, n°4, p. 422–434. < <http://dx.doi.org/10.1016/j.landurbplan.2005.12.003> >

Johansson B., Cipolla, 2002, A system for automatic pose-estimation from a single image in a city scene.

Jung C., Kim C., 2012, Real-time estimation of 3D scene geometry from a single image, *Pattern Recognition*, September 2012, vol. 45, n°9, p. 3256–3269. < <http://dx.doi.org/10.1016/j.patcog.2012.02.028> >

Karner K., Bauer J., Klaus A., Schindler K., 2002, Metropogis: A City Information System, 2002. < <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.9044> >

Khoo H.-K., Ong H.-C., Wong Y.-P., 2008, Image Texture Classification Using Combined Grey Level Co-Occurrence Probabilities and Support Vector Machines, in *Proceedings of the 2008 Fifth International Conference on Computer Graphics, Imaging and Visualisation*, Washington, DC, USA, IEEE Computer Society, p. 180–184. < <http://dx.doi.org/10.1109/CGIV.2008.47> >

Van Kreveld M., 1996, Variations on Sweep Algorithms: efficient computation of extended viewsheds and class intervals, *IN PROC. 7TH INT. SYMP. ON SPATIAL DATA HANDLING*, 1996, p. 13–15.

Lake I. R., Lovett A. A., Bateman I. J., Day B., 2000, Using GIS - and large-scale digital data to implement hedonic pricing studies, *International Journal of Geographical Information Science*, 2000, vol. 14, n°6, p. 521. < <http://dx.doi.org/10.1080/136588100415729> >

Li J., Taylor G., Kidner D., Ware M., 2006, Prediction of GPS Multipath Effect Using LiDAR Digital Surface Models and Building Footprints, in *Web and Wireless Geographical Information Systems*, p. 42–53. < [http://dx.doi.org/10.1007/11935148\\_5](http://dx.doi.org/10.1007/11935148_5) >

Liu L., Stamos I., 2005, Automatic 3D to 2D Registration for the Photorealistic Rendering of Urban Scenes, in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, Los Alamitos, CA, USA, IEEE Computer Society, p. 137–143. <<http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/CVPR.2005.80>>

Lowe D. G., 2004, Distinctive Image Features from Scale-Invariant Keypoints, *Int. J. Comput. Vision*, 2004, vol. 60, n°2, p. 91–110.

Luttik J., 2000, The value of trees, water and open space as reflected by house prices in the Netherlands, *Landscape and Urban Planning*, 1 May 2000, vol. 48, n°3-4, p. 161–167. < [http://dx.doi.org/10.1016/S0169-2046\(00\)00039-6](http://dx.doi.org/10.1016/S0169-2046(00)00039-6) >

Lynch K., 1976, *What Time is this Place?*, The MIT Press.

Microsoft H.-Y. S., Shum H., 1998, Interactive Construction of 3D Models from Panoramic Mosaics, PhD report, *IN PROC. CVPR*, 1998, p. 427–433.

Micusik B., Kosecka J., 2009, Piecewise planar city 3D modeling from street view panoramic sequences, in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, Los Alamitos, CA, USA, IEEE Computer Society, p. 2906–2912. < <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/CVPRW.2009.5206535> >

- Mikolajczyk K., Schmid C., 2005, A Performance Evaluation of Local Descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, vol. 27, n°10, p. 1615–1630.  
< <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/TPAMI.2005.188> >
- Miller D., 2001, A method for estimating changes in the visibility of land cover, *Landscape and Urban Planning*, 25 May 2001, vol. 54, p. 93–106. < [http://dx.doi.org/10.1016/S0169-2046\(01\)00128-1](http://dx.doi.org/10.1016/S0169-2046(01)00128-1) >
- Morello E., Ratti C., 2009, A digital image of the city: 3D isovists in Lynch's urban analysis, *Environment and Planning B: Planning and Design*, 2009, vol. 36, n°5, p. 837 – 853. < <http://dx.doi.org/10.1068/b34144t> >
- Ojala T., Pietikäinen M., Mäenpää T., 2002, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 2002, vol. 24, n°7, p. 971–987.
- Olufs S., Vincze M., 2011, Room-structure estimation in Manhattan-like environments from dense 2D range data using minimum entropy and histograms, in *2011 IEEE Workshop on Applications of Computer Vision (WACV), 2011 IEEE Workshop on Applications of Computer Vision (WACV)*, p. 118–124.  
< <http://dx.doi.org/10.1109/WACV.2011.5711492> >
- PUTRA S. Y., YANG P. P.-J., 2005, Analysing Mental Geography of Residential Environment in Singapore using GIS-based 3D Visibility Analysis, *International Conference: Doing, thinking, feeling home: the mental geography of residential environments in Delft, The Netherlands*, 2005.
- PUTRA S. Y., 2005, *GIS-based 3D volumetric visibility analysis and spatial and temporal perceptions of urban space*, 2005.
- Pyysalo U., Oksanen J., Sarjakoski T., 2009, Viewshed analysis and visualization of landscape voxel models, *24th International Cartographic Conference, Santiago, Chile*, 15 November 2009.
- Rana S., 2006, Isovist analyst: an arcview extension for planning visual surveillance.  
< <http://eprints.ucl.ac.uk/2104/> >
- Reitmayr G., Drummond, 2006, Going out: Robust modelbased tracking for outdoor augmented reality, *PROCEEDINGS OF 5TH IEEE AND ACM INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY (ISMAR)*, 2006, p. 109–118.
- Robert S., 2011, Cartographier la visibilité de la mer pour la gestion d'un littoral: une expérimentation sur la Côte d'Azur, *Espace géographique*, 2011, vol. 40, n°3, p. 215–230.
- Robertson D., Cipolla, 2004, An Image-Based System for Urban Navigation, *IN BMVC*, 2004, p. 819–828.
- Snavely N., Seitz S. M., Szeliski R., 2006, Photo Tourism: Exploring Photo Collections in 3D, *ACM TRANSACTIONS ON GRAPHICS*, 2006, vol. 25, p. 835–846.
- Sourimant, L. Morin, K. Bouatouch, 2009, *GPS, GIS AND VIDEO REGISTRATION FOR BUILDING RECONSTRUCTION*, Décembre 2009. < <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.149.2400> >
- Schmidhuber J., 2012, Multi-column deep neural networks for image classification, in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, IEEE Computer Society, p. 3642–3649. < <http://dl.acm.org/citation.cfm?id=2354409.2354694> >
- Teller S., Antone M., Bodnar Z., Bosse M., Coorg S., Jethwa M., Master N., 2003, Calibrated, Registered Images of an Extended Urban Area, *International Journal of Computer Vision*, Juin 2003, vol. 53, n°1, p. 93–107.  
< <http://dx.doi.org/10.1023/A:1023035826052> >

Turner A., Doxa M., O'Sullivan D., Penn A., 2001, From isovists to visibility graphs: a methodology for the analysis of architectural space, *Environ. Plann. B*, 2001, vol. 28, n°1, p. 103–121.  
< <http://dx.doi.org/10.1068/b2684> >

Xiaojuan N., WANG Y., 2013, 3D Reconstruction of Architecture Appearance: A Survey, in *JCIS Volume 9*, p. 3837–3848.

Yu-Chia Chung, Han T. X., Zhihai He, 2009, Building recognition using sketch-based representations and spectral graph matching, in *2009 IEEE 12th International Conference on Computer Vision, 2009 IEEE 12th International Conference on Computer Vision*, IEEE, p. 2014–2020. < <http://dx.doi.org/10.1109/ICCV.2009.5459444> >

Zhang, Kosecka J., 2006, Image Based Localization in Urban Environments, in *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, IEEE Computer Society, p. 33–40.

Zhang W., 2005, Localization based on building recognition, in *IEEE WORKSHOP ON APPLICATIONS FOR VISUALLY IMPAIRED*, 2005, vol. 3, p. 21–28.

## Références du chapitre 2

A. Le\_Bris,, N. Paparoditis, 2010, Matching terrestrial images captured by a nomad system to images of a reference database for pose estimation purpose, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS)*, 1 September 2010, vol. vol. 38, n°part 3A, p. 133–138.

Abidi M. a., Chandra T., 1995, A New Efficient and Direct Solution for Pose Estimation Using Quadrangular Targets: Algorithm and Evaluation, *IEEE Trans. Pattern Anal. Mach. Intell.*, 1995, vol. 17, n°5, p. 534–538.

Bay H., Ess A., Tuytelaars T., Van Gool L., 2008, Speeded-Up Robust Features (SURF), *Computer Vision and Image Understanding*, June 2008, vol. 110, n°3, p. 346–359. < <http://dx.doi.org/10.1016/j.cviu.2007.09.014> >

Bengio Y., 2009, Learning Deep Architectures for AI, *Found. Trends Mach. Learn.*, January 2009, vol. 2, n°1, p. 1–127. < <http://dx.doi.org/10.1561/22000000006> >

David P., Dementhon D., Duraiswami R., Samet H., 2004, SoftPOSIT: Simultaneous Pose and Correspondence Determination, *International Journal of Computer Vision*, 2004, vol. 59, n°3.  
< <http://portal.acm.org/citation.cfm?id=986707> >

Fischler M., Bolles R., 1981, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM*, June 1981, vol. 24, n°6, p. 381–395.

Haralick, Lee, Ottenberg, Nölle M., 1994, Review and analysis of solutions of the three point perspective pose estimation problem, *International Journal of Computer Vision*, 1 December 1994, vol. 13, n°3, p. 331–356.  
< <http://dx.doi.org/10.1007/BF02028352> >

Haralick, 1989, Determining camera parameters from the perspective projection of a rectangle, *Pattern Recogn.*, 1989, vol. 22, n°3, p. 225–230.

Hartley R., Zisserman A., 2004, *Multiple View Geometry in Computer Vision*, 2nd Revised ed. Cambridge University Press.

Holt R. J., Netravali A. N., 1991, Camera calibration problem: some new results, *CVGIP: Image Underst.*, 1991,

vol. 54, n°3, p. 368–383.

Lowe D. G., 2004, Distinctive Image Features from Scale-Invariant Keypoints, *Int. J. Comput. Vision*, 2004, vol. 60, n°2, p. 91–110.

Marquardt D., 1963, An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *SIAM Journal on Applied Mathematics*, 1963, vol. 11, n°2, p. 441, 431.

Mikolajczyk K., Schmid C., 2005, A Performance Evaluation of Local Descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, vol. 27, n°10, p. 1615–1630.

< <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/TPAMI.2005.188> >

Mikolajczyk K., Schmid C., 2004, Scale & Affine Invariant Interest Point Detectors, *International Journal of Computer Vision*, October 2004, vol. 60, n°1, p. 63–86.

< <http://dx.doi.org/10.1023/B:VISI.0000027790.02288.f2> >

Picard, D., Cord, M., Valle, E., 2009, Study of SIFT Descriptors for Image Matching based Localization in Urban Street View Context, 2009, n°CMRT09, p. 193–198.

Sourimant, L. Morin, K. Bouatouch, 2009, *GPS, GIS AND VIDEO REGISTRATION FOR BUILDING RECONSTRUCTION*, Décembre 2009. < <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.149.2400> >

Wolfe W. J., Mathis D., Sklair C. W., Magee M., 1991, The perspective view of three points, *IEEE Trans. Pattern Anal. Machine Intell.*, January 1991, vol. 13, n°1, p. 66–73. < <http://dx.doi.org/10.1109/34.67632> >

Yang Y., Cao Q., Lo C., Zhang Z., 2009, Pose Estimation Based on Four Coplanar Point Correspondences, in *Fuzzy Systems and Knowledge Discovery, Fourth International Conference on*, Los Alamitos, CA, USA, IEEE Computer Society, p. 410–414. < <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/FSKD.2009.310> >

Yuan, 1989, A General Photogrammetric Method for Determining Object Position, *RA*, 1989, vol. 5, n°2, p. 129–142.

## Références du chapitre 3

Anne Ruas, 1999, *Modèle de généralisation de données urbaines à base de contraintes et d'autonomie*, 1999.

Deering M., 1995, Geometry compression, in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM, p. 13–20. < <http://dx.doi.org/10.1145/218380.218391> >

Favier E., 1994, *Contribution de l'analyse multi-résolution à la description des contours et des textures*, PHD report, l'Université de Saint-Etienne.

Qingsheng G., Brandenberger C., Hurni L., 2002, A progressive line simplification algorithm, *Geo-spat. Inf. Sci.*, September 2002, vol. 5, n°3, p. 41–45. < <http://dx.doi.org/10.1007/BF02826387> >

Touma C., Gotsman C., 1998, Triangle Mesh Compression, *Graphics Interface 98 Conference Proceedings*, 1998, p. 26–34.

Turner A., Doxa M., O'Sullivan D., Penn A., 2001, From isovists to visibility graphs: a methodology for the analysis of architectural space, *Environ. Plann. B*, 2001, vol. 28, n°1, p. 103–121.  
< <http://dx.doi.org/10.1068/b2684> >

Yan L., Lianhe Y., 2009, Based on Delaunay Triangulation DEM of Terrain Model.  
< <http://ccsenet.org/journal/index.php/cis/article/view/1810> >

## Références du chapitre 4

Bentley, Ottmann, 1979, Algorithms for Reporting and Counting Geometric Intersections, *IEEE Transactions on Computers*, September 1979, vol. C-28, n°9, p. 643–647. < <http://dx.doi.org/10.1109/TC.1979.1675432> >

Bilsen V., 2010, 3D Visibility Analysis in Virtual Learning Environments and Interactive and Digital Media, in *Interactive & Digital Media for Education in Virtual Learning Environment*, New York.

Do E. Y.-L., 1997, Tools for visual and spatial analysis of CAD models, *CAAD Futures 1997 Conference*, 4 August 1997, p. 373–388.

Fisher-Gewirtzman D., Shach Pinsky D., Wagner I. A., Burt M., 2005, View-oriented three-dimensional visual analysis models for the urban environment, *Urban Design International*, April 2005, vol. 10, p. 23–37.  
< <http://dx.doi.org/10.1057/palgrave.udi.9000133> >

Jia W., Yi W.-J., Saniie J., Oruklu E., 2012, 3D image reconstruction and human body tracking using stereo vision and Kinect technology, in *2012 IEEE International Conference on Electro/Information Technology (EIT), 2012 IEEE International Conference on Electro/Information Technology (EIT)*, p. 1–4.  
< <http://dx.doi.org/10.1109/EIT.2012.6220732> >

Rana S., Batty M., 2004, Visualising the structure of architectural open spaces based on shape analysis,  
< <http://eprints.ucl.ac.uk/187/> >

Rana S., 2006, Isovist analyst: an arcview extension for planning visual surveillance.  
< <http://eprints.ucl.ac.uk/2104/> >

Rembliez F., 2011, *Tourner en 3D-relief: de la pré-production à la diffusion*, Paris, Eyrolles,  
Schmidhuber J., 2012, Multi-column deep neural networks for image classification, in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, IEEE Computer Society, p. 3642–3649. < <http://dl.acm.org/citation.cfm?id=2354409.2354694> >

Suleiman W., Joliveau T., Favier E., 2011, 3D Urban Visibility Analysis with Vector GIS Data, in *GISRUK 2011*, University of Portsmouth, UK, 27-29 avril 2011. < [http://portail.univ-st-etienne.fr/servlet/com.univ.collaboratif.utils.LectureFichier?CODE\\_FICHER=1315906066024&ID\\_FICHE=185671](http://portail.univ-st-etienne.fr/servlet/com.univ.collaboratif.utils.LectureFichier?CODE_FICHER=1315906066024&ID_FICHE=185671) >



## Publications scientifiques

- SULEIMAN W., JOLIVEAU T. & FAVIER E., 2012 - *Une nouvelle méthode de calcul d'isovist en 2 et 3 dimensions*. Conférence SAGEO 2012, 6-9 novembre 2012, Liège, Belgique.
- SULEIMAN W., JOLIVEAU T. & FAVIER E., 2012 - *A New Algorithm for 3D Isovist*. In 15th International Symposium on Spatial Data Handling Geospatial dynamics, geosimulation and exploratory visualization, 22-24th August 2012 in Bonn, Germany. Advances in Spatial Data Handling Advances in Geographic Information Science 2013, pp 157-173.
- SULEIMAN W., JOLIVEAU T. & FAVIER E., 2012 - *Reconnaissance de bâtiments et localisation de photographies au moyen d'un descripteur de texture*. In Revue Internationale de Géomatique, Vol. 22/3 - 2012 - pp.439-459 - doi:10.3166/ri.22.439-459.
- SULEIMAN W., JOLIVEAU, T. & FAVIER, E., 2011 - *Buildings Recognition and Camera Localization Using Image Texture Description*. Conférence SAGEO, 3-8 Juillet 2011, Paris, France.
- SULEIMAN W., JOLIVEAU T. & FAVIER E., 2011 - *3D Urban Visibility Analysis with Vector GIS Data*. In GISRUK, 27-29th April 2011, University of Portsmouth, UK.