

ZFOREST : UN PROTOTYPE DE PLATEFORME WEB DE COVISUALISATION LIDAR, RASTER ET VECTEUR À GRANDE ÉCHELLE.

David Vandergucht¹, Nicolas Paparoditis¹, Alexandre Devaux¹, Mathieu Bénard¹, Clément Mallet¹

1: Univ. Paris-Est, IGN, MATIS, F-94160

Résumé

En recherche comme en exploitation forestière, la donnée Lidar aéroportée apporte des clefs de compréhension sur la structure du terrain, de la forêt et par extension, des informations sur la biomasse aérienne. Mais pour être utile, cette donnée Lidar doit couvrir de vastes étendues tout en étant très résolue spatialement. Ces caractéristiques se traduisent par de grands volumes de données très difficiles à visualiser, manipuler et étudier sans l'aide de logiciels très onéreux. Dans le cadre du projet ANR FORESEE, nous avons développé un logiciel web, de visualisation mixte nuage de points Lidar / surface 3D issue d'un Modèle Numérique de Terrain / carte / photographie aérienne et terrestre / donnée vectorielle : la plateforme zForest. Ce logiciel, qui s'adresse aux chercheurs en télédétection et à terme aux exploitants forestiers, permet la navigation à grande échelle dans des données massives et leur exploration, du niveau de détail le plus large (la région) jusqu'au plus fin (l'arbre). Cet outil permet la mesure, l'annotation et l'extraction des données. Il propose également une interface de programmation web (API) permettant à d'autres outils du marché d'utiliser ses données sources. zForest étant une plateforme web, elle est disponible sans installation sur tous les navigateurs internet récents, facilitant son accessibilité et son déploiement.

Mots clés : lidar aéroporté, WebGL, cartographie grande échelle, forêts, stockage de données massives lidar, quadtree, visualisation de données, 3D

Abstract

For research purpose as well as forestry, airborne Lidar data provides insights to understand the terrain morphology, the 3D forest structure and, by extension, information on aerial biomass. But to be useful, the Lidar data must cover a very wide area while providing a very high spatial resolution. Those features translate into massive amounts of data that are very hard to display, handle and study without the help of expensive dedicated softwares. We have developed a rich internet application to provide hybrid Lidar pointcloud, 3D Digital Terrain Model, raster and vector data visualization in the web browser : zForest. This software, which targets researchers in remote sensing as well as foresters, enables the user to navigate and explore a massive amount of large scale airborne Lidar data, from a wide point of view (a French administrative departement) to a very narrow one (a tree). This application provides measuring tools, annotation tools and tools for data extraction. The data extraction part is available through a web application programming interface for other (external) tools to harvest the raw data used in zForest. Being a web platform, zForest can be easily loaded in any recent web browser and does not need any extra software, thus making it easy to access and deploy.

Keywords : Airborne Lidar, WebGL, large scale mapping, mountains areas, forest, lidar data storage, quadtree, data visualization, 3D

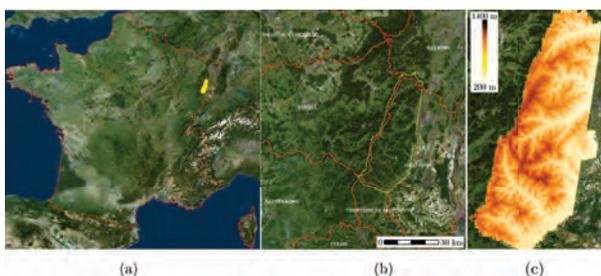
1. Contexte

L'utilisation des données Lidar en complément des techniques de relevés traditionnels fournit aux chercheurs comme aux forestiers une nouvelle manière d'examiner la forêt en trois dimensions. Le Lidar aéroporté apporte en effet une information volumétrique, dévoilant les structures végétales habituellement cachées par la canopée aux yeux des capteurs permettant d'acquérir des images satellitaires, aériennes, comme les prises de vue effectuées par l'Institut National de l'Information Géographique et Forestière (IGN), ou encore par drones.

Il permet de mieux comprendre la nature et la composition du peuplement caché par les arbres dominants, en croisant le nuage de points généré par le Lidar avec

les relevés dendrométriques de terrain. En outre, sa méthode d'acquisition (levé aéroporté) permet de couvrir de très grandes surfaces avec une précision très élevée (<40 cm en altimétrie sous couvert végétal selon les conditions d'acquisitions, la densité de couvert et la topographie du terrain), et d'obtenir une quantité d'information inaccessible par relevés sur le terrain.

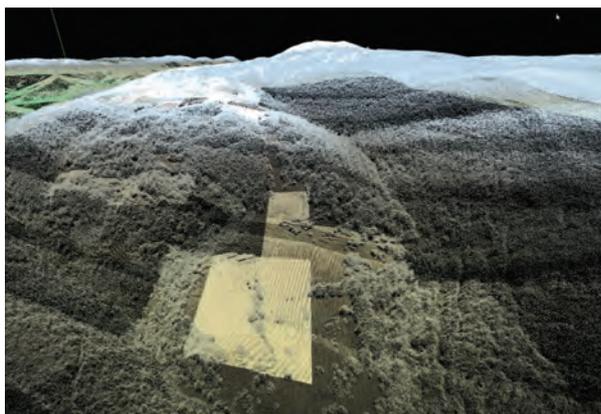
Pour les exploitants forestiers, le Lidar apporte une information très concrète sur l'aspect du terrain et les conditions de mobilisation et d'exploitation du bois. Pour autant, les données Lidar brutes ne sont pas d'une utilisation facile ou grand public. Leur lecture et leur traitement nécessitent des outils professionnels parfois coûteux, leur interprétation est obscure pour un non-initié



(1) Zone de test du projet Foresee : 1300 km² dans les Vosges.



(2) Vue générale du nuage de points Lidar de la zone de test. Le nuage contient 5,3 milliards de points et occupe 190Go.



(3) Vue rapprochée sur le Grand Ballon, point culminant du massif des Vosges

FIGURE 1: Localisation de la zone d'étude des Vosges (ANR FORESEE) et nuage de points correspondant à cette zone dans le logiciel iTowns / zForest. Le nuage de points est coloré en fonction de l'altitude (vert en bas, blanc en haut) et de l'information d'intensité réfléchie des points (plus ou moins sombre).

sans l'apport d'informations complémentaires sur lesquelles s'appuyer, comme par exemple la photographie aérienne, ou les produits que l'on peut en dériver (cartes de hauteur dominante, surface terrière, etc.).

Enfin, les grandes surfaces levées par l'acquisition aérienne ont pour défaut d'être très difficiles (voire impossibles) à manipuler de manière interactive en raison du volume des données.

En résumé, cette donnée qui est de plus en plus présente, gratuite dans certains pays, et qui permet tant de nouveaux développements, est loin d'être aussi simple d'accès que l'information géographique traditionnelle aujourd'hui.

Dans le cadre du projet ANR FORESEE, nous avons souhaité proposer un logiciel orienté vers la recherche et l'exploitation forestière qui permette d'explorer et de travailler facilement sur ces données : zForest. Ce logiciel a été développé pour visualiser les acquisitions Lidar à grandes échelles réalisées par l'IGN dans le cadre de ce projet, sur plus de 1300 km² dans les Vosges (voir Figure 1). Cette zone de test servait par ailleurs à l'application des méthodes de télédétection développées sur le projet.

La partie 2 de cet article donne une vue générale des caractéristiques de cet outil, notamment les particularités liées à son environnement d'utilisation. Celui-ci constitue une contrainte spécifique, au même titre que la nature et le volume des données exploitées par l'outil. La partie 3 explicite ces contraintes, et la partie 4 détaille l'architecture mise en place pour réaliser une solution de visualisation Lidar grande échelle répondant à ces exigences. Il s'agit de la partie la plus technique de cet article. Comme évoqué plus haut, zForest doit permettre la visualisation combinée de Lidar avec d'autres sources d'informations. La partie 5 recense les différents outils et systèmes de visualisations qui peuvent être combinés aux nuages de points Lidar. Enfin la partie 6 rassemble ces idées et propose quelques perspectives sur l'évolution et l'utilisation de cet outil.

2. Le logiciel web zForest

zForest est un logiciel « web » qui permet de visualiser des données géographiques composites. Il s'agit d'une évolution d'un outil IGN préexistant, iTowns (Devaux et Paparoditis, 2010; Devaux et al., 2012) (voir Figure 2), qui cible plus particulièrement la visualisation et l'exploration des nuages Lidar acquis en milieu urbain par le véhicule Stéréopolis de l'IGN (Paparoditis et al., 2012). zForest propose un ensemble de fonctionnalités complémentaires pour adapter iTowns à d'autres types de données et d'environnements de navigation.

L'outil permet en effet l'affichage d'un Modèle Numérique de Terrain, de couches cartographiques diverses, de données vectorielles, de photographies géolocalisées



FIGURE 2: L'application web iTowns permet de naviguer dans des nuages de points Lidar urbains et des images panoramiques, à l'aide d'un navigateur internet, sans plug-in additionnel.

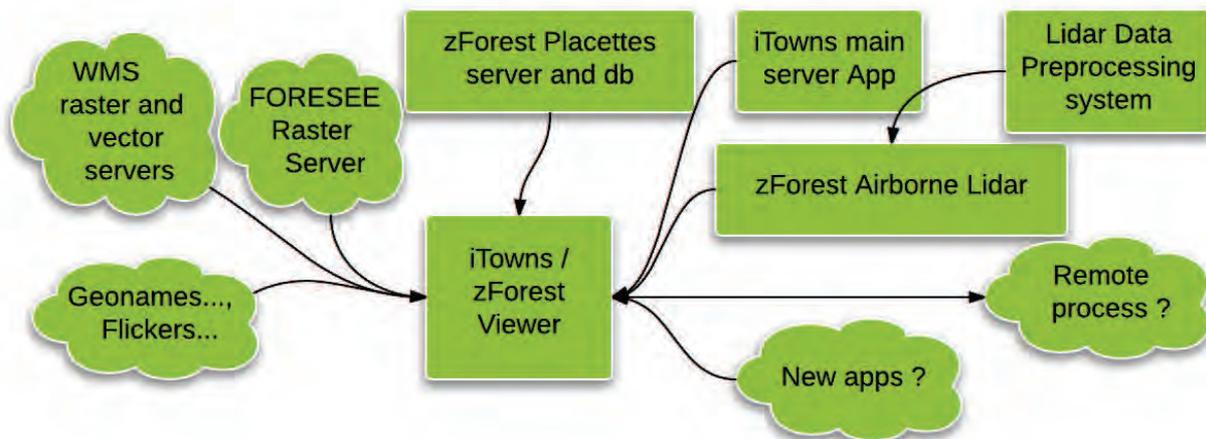


FIGURE 3: Architecture générale de zForest et ouverture vers des services tiers. L'application est composée d'un client riche (viewer) ainsi que de plusieurs serveurs dédiés : le serveur iTowns fournit un certain nombre de services d'accès aux bases de données, le serveur de nuage de points Lidar dédié à zForest ainsi que la suite logicielle permettant de bâtir le modèle de données, et enfin, des points de connexions vers des serveurs tiers dont les données sont exploitées par zForest comme des serveurs d'images WMS.

et surtout, du nuage de points Lidar, le tout à très grande échelle.

Ce logiciel est développé en utilisant les standards du web 2.0 à savoir html5, JavaScript et surtout WebGL, une technologie d'affichage 3D qui ne nécessite pas l'installation de programmes tiers pour fonctionner (à la différence de Adobe Flash, par exemple). Il suffit d'utiliser un navigateur internet récent qui intègre normalement ces fonctionnalités en standard. Ces choix technologiques confèrent à iTowns et zForest une grande accessibilité, car il n'y a pas besoin d'installer d'application lourde comme Terrascan (2002); CloudCompare (2006) ou FugroViewer (2011). Le logiciel fonctionne sur Windows, OSX et Linux.

zForest propose un système de visualisation multi échelles qui permet de naviguer dans la donnée Lidar du point de vue le plus général au point de vue le plus détaillé. Cette fonctionnalité rend possible la navigation continue dans un nuage de points de très grande dimension, car les données sont chargées à la demande en fonction du point de vue et du niveau de précision nécessaire. Les données superflues ne sont pas affichées. Notre approche repose donc sur une préparation des données et un fonctionnement applicatif orienté « gestion du niveau de détail ».

Cet outil permet à l'utilisateur d'identifier facilement des zones d'intérêt par l'exploration interactive et « visuelle » des données, et permet d'effectuer plusieurs actions sur ces données. L'application propose des outils de mesure ainsi que la possibilité d'exporter des zones pointées par les utilisateurs pour un travail dans des logiciels dédiés.

Ce système d'extraction est, de plus, exposé sous la forme d'un webservice qui répondra à terme au standard WPS (Web Processing Service). Les données internes proposées par zForest pourront donc être exploitées di-

rectement par des outils tiers et cet élément constitue une brique vers la construction d'un écosystème d'outils dédié au travail sur le Lidar forestier (voir Figure 3).

3. Contraintes techniques

Nous souhaitons pouvoir explorer le contenu d'un nuage de points de grande dimension (70 km x 20 km, 5,3 milliards de points, 190 Go) de manière continue. Ce nuage de points est le fruit d'un scan aérien effectué par l'IGN dans les Vosges dans le cadre du projet ANR FORESEE (voir Ferraz et al. (2015) pour plus de précisions sur la zone et les paramètres d'acquisition). Cette exploration doit pouvoir se faire en ligne, sur une machine ne disposant pas des données Lidar localement, ni de logiciel dédié de visualisation Lidar.

Nous avons trois types de contraintes associées à ces postulats de départ :

- Affichage : il n'est pas possible d'afficher un nombre de points aussi grand dans les cartes graphiques des machines actuelles pour des raisons de taille mémoire (512 Mo pour une Nvidia GT218).
- Taille mémoire : la mémoire RAM d'un ordinateur classique (4 ou 8 Go) ne peut stocker l'intégralité des données. Il faut sélectionner les données pertinentes à afficher et gérer cet espace (même problème que pour la carte graphique).
- Temps de transfert : le temps nécessaire pour transférer les 190 Go de données avec les connexions internet usuelles est prohibitif.
- Espace de stockage : personne ne souhaite stocker l'intégralité des données Lidar sur une machine locale, même sur disque dur.

Tous ces points soulignent l'importance de la mise en place d'une stratégie ne nécessitant pas le chargement complet des données pour les visualiser.



FIGURE 4: Synopsis général du fonctionnement de zForest. L'utilisateur déplace une caméra 3D dans l'application. Un système de partitionnement de l'espace détecte quelles sont les zones visibles et à quel niveau de détail. Des demandes de récupération de données Lidar sont émises vers un gestionnaire de requêtes qui interroge à son tour un serveur sur lequel les données sont stockées d'une manière optimisée. Quand les données Lidar sont récupérées par le client, elles sont décodées et affichées à l'écran.

4. Architecture générale

Le fonctionnement de notre plateforme s'articule autour de trois grands ensembles (voir Figure 4) :

- Une structure de données optimisée.
- Un système de découpage de l'espace en fonction de la position de l'utilisateur.
- Un système de gestion des transferts de données par file d'attente.

Notre approche s'appuie sur les structures de stockage bien connues que sont les arbres récursifs, en 2 (quadtree) ou 3 dimensions (octree). Le développement de cette plateforme s'inspire de plusieurs travaux, notamment Scheiblauer (2008); Elseberg et al. (2013) pour ce qui est des structures de stockage de nuage de points. Les travaux de l'université de Vienne ont donné lieu à l'outil Scanopy, lui-même à l'origine de Potree (2012), un logiciel WebGL utilisé pour la visualisation de scans Lidar architecturaux très denses, et qui est également un axe d'inspiration très important pour nous.

Cependant, notre donnée Lidar est issue de scans aériens. Il s'agit d'une information dont les dimensions

sont beaucoup plus étendues en planimétrie qu'en altimétrie (70 km x 20 km x 600 m). Pour cette raison, nous avons préféré considérer cette information comme 2 D ou 2,5 D plutôt que 3 D, et nous appuyer sur un quadtree modifié plutôt que sur un octree (voir Figure 5).

L'utilisation du quadtree a d'autres bénéfices pour nous, car nous avons également besoin d'afficher un Modèle Numérique de Terrain (MNT). Le MNT est livré sous la forme de grilles d'élévations, qui sont stockées sous forme d'images... dans un quadtree. Pour le MNT, nous sommes inspirés des travaux de Dick et al. (2009); Wiendl et Dick (2013); Cellier (2014).

4.1. Structuration des données

Les données Lidar sont préparées en amont par une méthode qui s'apparente au tuilage et au stockage pyramidal utilisé pour naviguer dans de très grandes images. Le processus de préparation des données prend un ensemble de fichiers Lidar en entrée (format LAS) et produit une structure de fichiers pyramidale, contenant également des fichiers Lidar au format LAS.

Une telle structure permet de gérer le niveau de détail et fournit des méthodes de recherche performantes dans les données car chaque "étage" de la pyramide héberge une version de plus en plus détaillée de l'information. La « base » de la pyramide (aussi appelée « feuilles ») présente un ensemble de cellules contenant l'intégralité de l'information la plus détaillée de notre base de données.

En remontant dans les étages vers le haut de la pyramide (la racine de l'arbre), cette information est sous échantillonnée afin de présenter un niveau de détail moindre, donc plus léger. De la racine aux feuilles, chaque cellule est subdivisée en quatre cellules filles, chaque niveau contient donc quatre fois plus d'information que le précédent. C'est la raison pour laquelle on utilise le terme « quadtree ». En lecture, le choix du niveau de détail se fait donc en choisissant un niveau de profondeur de l'arbre et en y lisant les fichiers qui s'y trouvent.

Selon leur position dans l'arbre de stockage, les nouveaux fichiers Lidar sont :

- Des extractions non modifiées de portions de fichiers Lidar d'origine (feuilles).
- Des portions agrégées et sous-échantillonnées des branches situées sous eux (niveaux intermédiaires).

Les fichiers stockés dans la structure ne contiennent donc pas de données interpolées.

4.2. Réalisation de la structure de données

Le processus de création du quadtree fonctionne en trois temps (voir Figure 6) :

1. Détection des paramètres de découpage par analyse des boîtes englobantes des fichiers Lidar. Ces paramètres peuvent également être forcés par le développeur.

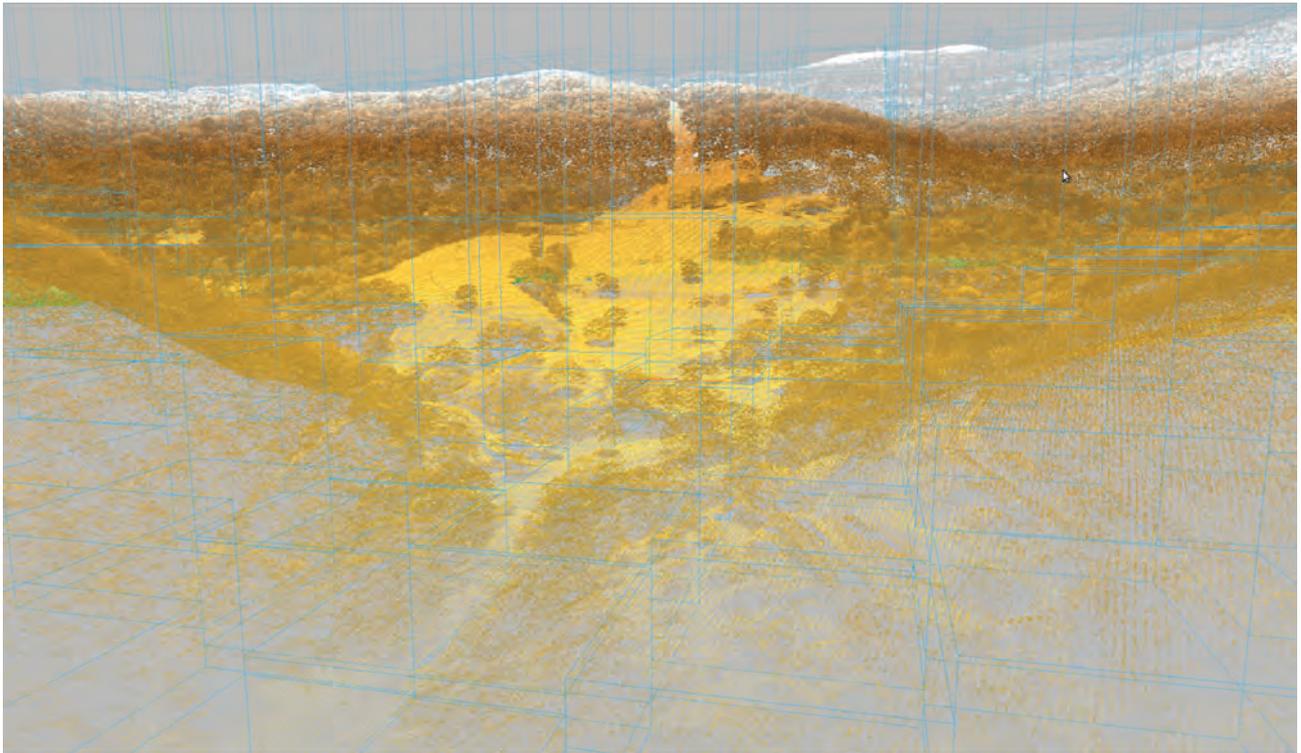


FIGURE 5: Une partie du nuage de points Lidar des Vosges, visualisée dans iTowns / zForest. Les boîtes englobantes de chacun des contenus Lidar associés avec une des cellules du quadtree sont affichées en bleu. Les points sont colorés en fonction de l'altitude et de l'attribut d'intensité. Le nuage de points Lidar est suffisamment précis pour voir la forme des arbres. La nature même du Lidar permet d'obtenir des informations sur l'intérieur de la canopée, souvent jusqu'au sol sous l'arbre.

2. Découpage récursif des fichiers jusqu'à atteindre un critère d'arrêt, et création de la structure pyramidale de stockage. Dans notre cas, le critère d'arrêt est l'atteinte d'un nombre maximal de points par fichier unique.
3. Remontée depuis les feuilles de l'arbre vers la racine par agrégation des fichiers situés au niveau précédent, et suppression de trois points sur quatre.

Cet outil a été implémenté en python et utilise les exécutables LasTools (2006) pour découper, ré-agréger et ré-échantillonner les fichiers. Le script python orchestre la création de la structure, pilote l'exécutable, et fournit une lecture basique des fichiers Lidar binaires (notamment leurs boîtes englobantes). La Table 1 donne une idée des temps de création de quadtrees sur plusieurs jeux de données. Ce temps varie dramatiquement selon la configuration des disques durs de la machine utilisée, car le processus repose de manière importante sur des opérations de lecture et écriture. La structure de fichier générée est ensuite exploitée par le serveur de données.

4.3. Serveur de données

Le serveur de données utilisé dans le cadre de ce démonstrateur est un serveur Node.js (2009). Node.js est un ensemble d'outils permettant d'utiliser le langage JavaScript dans un interpréteur indépendant du navigateur internet, là où l'on exécute JavaScript habituellement.

Site	orig (Go)	qtree (Go)	nb pts (Gpts)	tps (min)
Vosges	137	190	5,3	1940
Aillons	11,6	15	0,417	121
Bauges	35,4	45	1,2	85
Ventoux	3,3	4,2	0,12	10

TABLE 1: Volume d'origine des fichiers Lidar en Gigaoctets, volume du quadtree généré, nombre de points en milliards et temps de création du quadtree en minutes pour quelques sites de test du projet ANR FORESEE.

Il est donc possible d'utiliser JavaScript de manière similaire à d'autres langages de scripts comme Python ou PHP. Node.js permet notamment de créer facilement un serveur web fonctionnant de manière asynchrone et événementielle.

Ce type de fonctionnement est particulièrement avantageux dans notre cas de figure, car le fonctionnement de l'application finale demande énormément d'accès disque. La nature asynchrone de Node.js permet de répondre simultanément à plusieurs demandes de fichiers Lidar de manière non bloquante pour l'application (le serveur ne se bloque pas pendant qu'il va lire un fichier sur le disque et accepte d'autres requêtes dans le même temps).

De plus, Node.js permet la réutilisation du code JavaScript développé pour le côté client de l'application, du côté du serveur. Cette stratégie est notamment utilisée

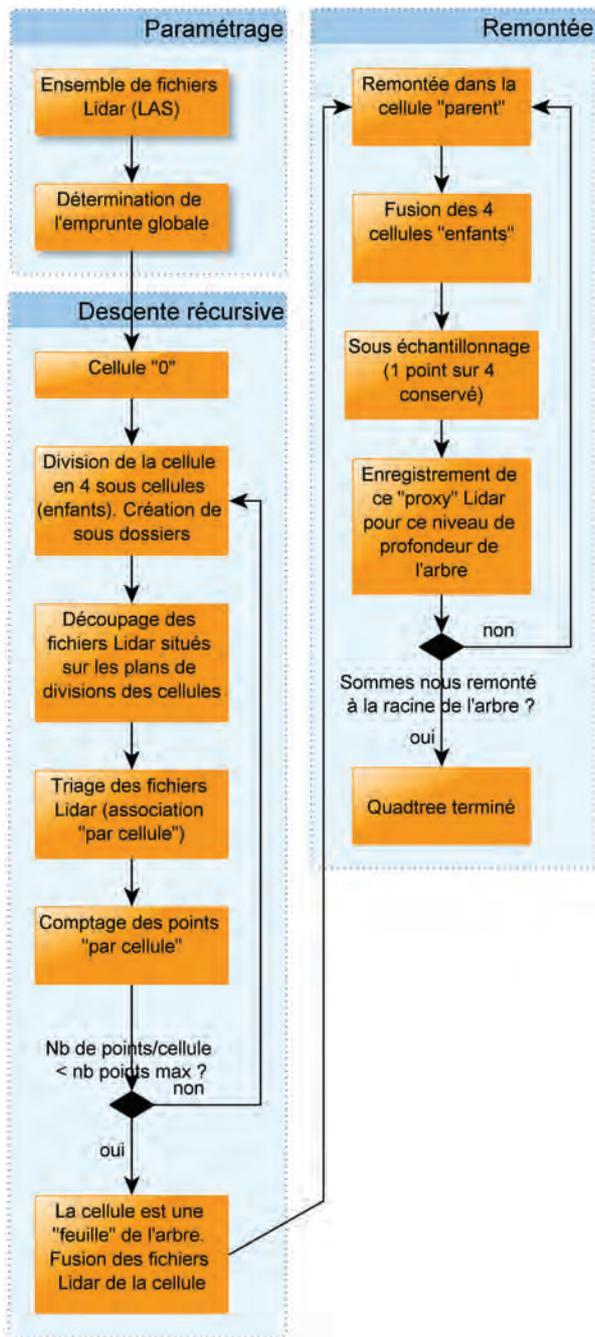


FIGURE 6: Processus de création récursif du quadtree. Les fichiers Lidar d'origine sont d'abord scannés pour connaître la dimension totale de la zone, puis la récursion commence, en alternant des phases de découpe de fichier Lidar et de comptage des points par cellule. La récursion s'arrête quand les fichiers Lidar des feuilles ne contiennent pas plus de 65536 points. Le processus passe alors en phase de remontée : il construit les niveaux de détail successifs de l'arbre, des feuilles, jusqu'à la racine.

dans le cadre de la mise en place du service de récupération de données Lidar à la demande (voir section 5.5), qui utilise les mêmes organes que la partie client.

L'accès aux fichiers Lidar à travers node.js se fait de manière très simple : une requête http « get » permet de spécifier le niveau de profondeur de l'arbre demandé

et le numéro de tuile approprié. Cette approche s'apparente dans l'idée au WMTS (Web Map Tile Service) qui est un protocole de requête standardisé permettant de demander à un serveur l'envoi de données de type raster (image) géolocalisée et tuilée. Néanmoins, nous ne suivons pas encore le formalisme WMTS défini par l'Open Geospatial Consortium (OGC), organisme chargé de la standardisation des interfaces en lien avec l'information géographique.

En cas de requête trop « profonde » (niveau de profondeur supérieur au niveau de profondeur proposée par le serveur), le serveur retourne un code HTML spécifique permettant de signaler au client qu'il n'y a pas de donnée plus fine, et qu'il faut conserver la meilleure donnée dont il dispose pour cette branche.

4.4. Client de visualisation

Le côté « client », la partie de l'application exécutée dans le navigateur internet de l'utilisateur, s'appuie sur la plateforme iTowns développée par le laboratoire Matis de l'IGN.

Cette plateforme comporte un ensemble d'outils 3D développés en JavaScript autour de la librairie Three.js (2010), permettant de calculer des géométries 3D dans le navigateur internet en utilisant la technologie WebGL.

Dans zForest, le chargement des données et le choix du niveau de détail doivent se faire de manière automatique en fonction des déplacements de la caméra 3D pilotée par l'utilisateur.

Le chargement des données doit également se faire de manière continue, asynchrone et non bloquant, car cette opération ne doit pas empêcher la navigation de l'utilisateur. Pendant le chargement des données, l'application doit, autant que possible, montrer des données de remplacement moins détaillées permettant de guider la navigation.

Ces contraintes conduisent au développement de trois organes (voir Figure 7) :

- Système de découpe de l'espace et d'analyse du niveau de détail asynchrone ;
- Système de requête ;
- Système de cache et de gestion des ancêtres ;

4.4.1. Système de découpe de l'espace et d'analyse du niveau de détail asynchrone

Ce système a pour fonction d'estimer à tout instant, la meilleure configuration d'affichage de la donnée Lidar en fonction d'un point de vue particulier, c'est-à-dire, quelles sont les tuiles de la structure de données vues à cet instant, et quel niveau de profondeur de l'arbre utiliser (niveau de détail).

L'ensemble de la zone à afficher est représenté du côté du client par une boîte englobante, qui va être subdivisée en sous-cellules comme nous l'avons présenté précédemment pour la préparation des données. L'axe vertical n'est pas considéré dans la subdivision. Ici, le critère d'arrêt est la taille d'une sous-cellule projetée dans la vue caméra : si la cellule est supérieure à 256 pixels, elle est subdivisée. La taille projetée est fonction de la

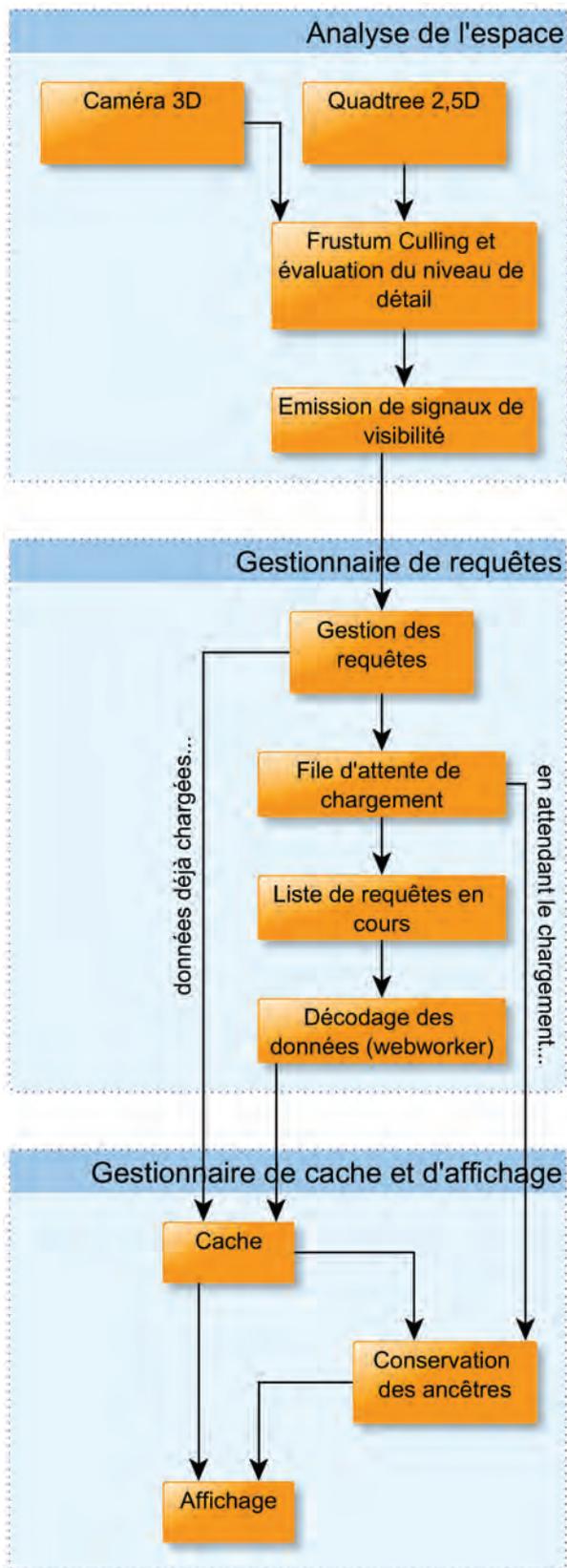


FIGURE 7: Organes principaux du client de visualisation.

taille absolue de la sous-cellule, de la distance à la caméra et de la focale utilisée, on comprend donc de manière intuitive que les zones éloignées apparaissant petites dans l'image sont peu subdivisées et le niveau de

détail associé reste faible (voir Figure 8). De plus, les cellules hors du cadre de la caméra (frustum) ne sont pas traitées.

Ce processus de découpage de l'espace se fait indépendamment des données, sur un ensemble de boîtes vides. La subdivision n'est donc pas associée à un traitement lourd et l'analyse de l'espace se fait en temps réel. Le seul objectif de ce système est de signaler à un autre système (voir section 4.4.2), quelles sont les données à charger et à quel niveau de détail. Ce signalement est opéré par l'émission de signaux de visibilité, comme le montre la Figure 9.

La visibilité des cellules est ainsi évaluée 60 fois par seconde, et ce, uniquement quand la caméra est en mouvement. Le système de quadtree permet de limiter le nombre de tests de visibilité (400 cellules « valides » pour le critère d'arrêt évoqué et une résolution d'écran de 1920x1080 pixels). Le positionnement de marqueurs dans les cellules de l'arbre permet d'identifier les changements dans la structure d'une évaluation sur l'autre. Lors d'un mouvement de caméra « standard » et à 60 images par seconde, 80 à 90% de la structure reste identique entre deux images. Le marquage permet d'éviter l'émission de signaux redondants de visibilité, en ne se concentrant que sur les différences de structure.

Ce système de quadtree « émettant des signaux » fonctionne donc de manière relativement autonome : d'autres processus écoutent les signaux d'affichage (visibilité) ou de masquage des cellules qui le compose, mais le quadtree n'a aucune connaissance du type de donnée qu'il sert à afficher. Il se contente de partitionner l'espace. Un même objet peut envoyer des signaux à plusieurs fournisseurs de donnée (Lidar, Images, vecteurs) dès lors que leurs paramètres de requêtes sont compatibles avec les dimensions et les sous-divisions du quadtree.

4.4.2. Système de requête et système de cache

Chaque signal émis par le quadtree comporte un identifiant unique de cellule permettant de connaître sa position dans l'arbre (sa « profondeur » et son numéro de tuile pour ce niveau de profondeur). Il comporte également des informations concernant la taille géométrique de cette cellule.

Ces signaux sont écoutés par un gestionnaire de tuile Lidar (LasManager). Ce gestionnaire se charge de collecter les besoins en tuile annoncés par le quadtree, d'organiser les requêtes, le décodage des fichiers reçus et le placement dans un cache des données décodées. Il s'occupe également d'afficher ces données, si celles-ci sont toujours requises.

4.4.3. Gestion des requêtes

Lors de la navigation, l'état de visibilité des cellules de l'arbre change en permanence à un rythme soutenu, indépendant du temps de traitement des données associées. En raison de leur poids et de la vitesse de la connexion internet de l'utilisateur et du serveur, les données Lidar mettent un certain temps à arriver sur la machine. À ce temps de transfert, il faut ajouter un temps de

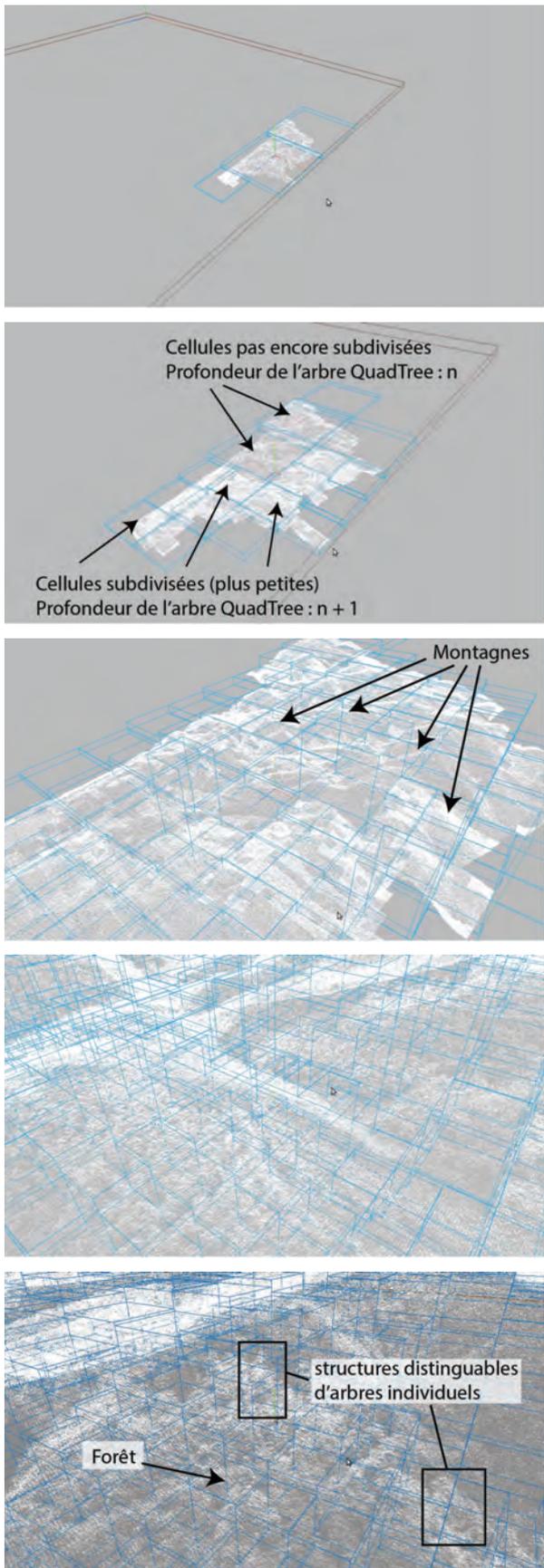


FIGURE 8: Vue du quadtree, dans le client de visualisation web. Si les cellules semblent « visuellement » de mêmes tailles, elles sont subdivisées en fonction du point de vue pour afficher des portions de nuage de points de plus en plus précises.

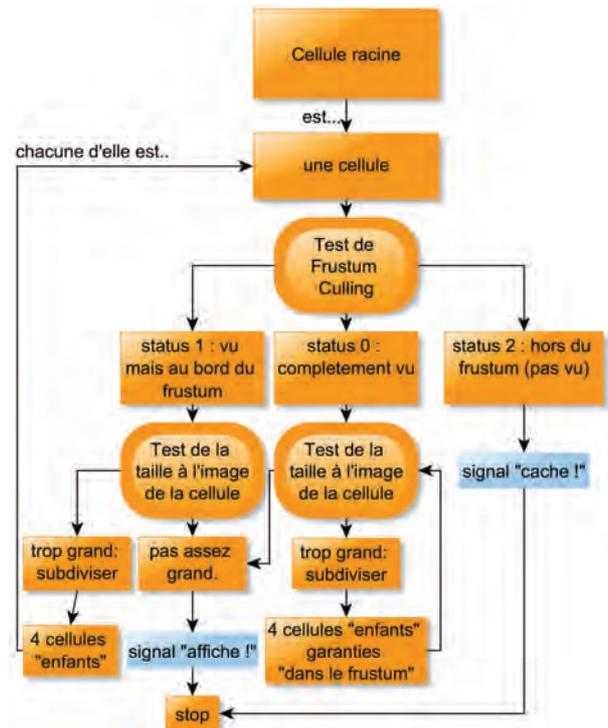


FIGURE 9: Schéma d'évaluation récursif du quadtree dans le client web, par intersection avec le frustum caméra et évaluation de la taille « vue » des cellules comme critère de subdivision.

décodage de fichier et de transfert des données vers la carte graphique. Il y a donc une latence importante entre l'arrivée des informations de visibilité ou de non-visibilité et l'affichage des données correspondantes.

Pour gérer cette latence, le LasManager propose plusieurs mécanismes : une file d'attente de chargement et un cache. Lorsque les signaux de visibilité de cellules sont reçus par le LasManager, celui-ci crée des ordres de requêtes pour des fichiers Lidar et les stocke dans une file d'attente. Un agent se charge alors de traiter cette file d'attente, à raison de 10 requêtes en simultané. Plusieurs « ordres de requêtes » sont donc extraits de la file d'attente et placés dans un espace de traitement où les requêtes sont exécutées. Lorsque la donnée correspondant à une requête a été récupérée et traitée, une place est libérée dans l'espace de traitement et une autre requête est extraite de la file d'attente et traitée à son tour.

La requête dont le traitement est terminé est alors dirigée, soit vers un espace d'affichage si elle est toujours utilisable (donc ces points partent vers la carte graphique), soit vers un espace de stockage si elle n'est plus nécessaire. Ce cas de figure arrive notamment si la caméra a bougé massivement pendant la phase de décodage de la donnée Lidar : la cellule à l'origine de l'ordre de requête a été marquée invisible pendant son traitement, mais la donnée a été décodée quand même. Elle n'est plus affichable, mais peut resservir donc on ne la jette pas pour autant.

Lorsqu'un nouvel ordre de requête est ajouté à la file d'attente, on commence par vérifier dans le cache si le contenu n'est pas déjà disponible en raison d'une de-

mande précédente, aboutie, mais avortée à la dernière minute. Si c'est le cas, le contenu sort du cache pour aller dans la zone d'affichage, et on ne crée pas de nouvel ordre de requête.

En cas de signal de masquage de tuile, il est possible d'annuler les ordres de requête à plusieurs endroits dans la chaîne de traitement :

- Quand l'ordre de requête est en attente dans la file d'attente.
- Quand les données sont en cours de transfert.

4.4.4. Décodage du fichier LAS et affichage

Le décodage du fichier Lidar au format LAS est réalisé en JavaScript par une librairie réalisée pour l'occasion. Cette librairie est en théorie capable de décoder les LAS 1.4, mais sur le projet elle n'a été testée qu'avec du LAS 1.2. Le décodage d'un fichier LAS est exigeant en ressource machine, car la manière dont les données Lidar sont représentées dans le format LAS est entrelacée : on a pour chaque point les coordonnées X Y Z et d'autres attributs. Pour les afficher en WebGL, nous avons besoin de construire plusieurs tableaux contenant pour l'un, les informations XYZ de tous les points, pour l'autre les informations d'intensité de tous les points, le numéro d'écho, etc.

Toujours dans le souci de bloquer le moins possible l'interaction avec l'utilisateur, cette opération de décodage est déportée dans un fil d'exécution (thread) applicatif secondaire par le biais d'un WebWorker. Il s'agit là d'un mécanisme proposé par les navigateurs web pour utiliser plusieurs threads et donc paralléliser les traitements, sans que ceux-ci ne bloquent complètement l'application.

4.4.5. Conservation des ancêtres

Quand la caméra 3D effectue un zoom sur une zone, des données avec un niveau de détail plus fin sont demandées au système. Cependant, le temps que ces données soient disponibles, il est préférable de conserver temporairement une donnée moins précise à l'écran afin d'assurer une continuité d'affichage. Lors de la demande d'affichage d'une nouvelle tuile, le système de conservation des ancêtres va chercher dans le cache si une tuile "parente" de la tuile attendue ne pourrait pas être utilisée en remplacement temporaire. Le système s'appuie sur la nomenclature des tuiles pour identifier l'ancêtre le plus proche (s'il y en a un de disponible) et l'afficher à l'écran. Cet ancêtre est alors configuré pour "écouter" les signaux du gestionnaire de requête, et disparaît automatiquement quand les tuiles qu'il remplaçait ont toutes été chargées. Néanmoins, les tuiles "ancêtres" sont, par essence, inadaptées au niveau de détail demandé pour la zone affichée : leur densité de point est trop faible, ce qui se traduit par un espace entre les points importants et des "trous" à l'image. Pour pallier ce problème, nous introduisons un facteur correctif sur la taille des points, afin que ceux-ci apparaissent plus gros et combent le vide. Ce facteur est une puissance de deux qui dépend de la différence de niveau de profondeur entre les tuiles

attendues et l'ancêtre affiché. Ces deux techniques permettent de conserver une relative continuité d'affichage lors d'une navigation plongeante.

4.4.6. Affichage

Comme nous l'avons évoqué dans la section 4.4.4, les données proposées dans les fichiers LAS nécessitent un traitement préalable à leur affichage.

Nous sommes en effet confrontés à deux contraintes :

- Chaque point dans le fichier LAS contient des attributs qui ne sont pas forcément nécessaires à l'affichage (sauf s'ils sont explicitement demandés par l'utilisateur). Il faut pouvoir faire le tri en amont de la carte graphique pour optimiser l'espace mémoire.
- Les attributs sont de type incompatible avec les tableaux de valeurs actuellement acceptés par WebGL (uniquement tableaux de float).

Pour pallier le premier problème, nous désentrelaçons les points du fichier LAS et créons en RAM plusieurs tableaux de valeurs (ArrayBuffer JavaScript) contenant les attributs. Seuls les tableaux nécessaires à l'affichage sont copiés vers la mémoire de la carte graphique, ce qui correspond le plus souvent au tableau contenant les coordonnées XYZ. Pour les attributs ne correspondant pas au type float (par exemple des entiers sur 8 bits) nous groupons les attributs et les empaquetons sur des mots de 32 bits que nous faisons interpréter comme des floats par la carte graphique. Une fonction spécifique est alors utilisée dans le shader pour réinterpréter ces "faux" float selon leur type d'origine (par exemple 4 entiers sur 8 bits).

L'utilisateur choisit l'attribut à visualiser dans l'interface. Afin de n'envoyer que les tableaux nécessaires à la carte graphique ainsi que les fonctions de dépaquetage adéquates en fonction de la sélection d'attributs, zForest assemble et compile un shader GLSL dédié (OpenGL Shading Language - programme exécuté sur la carte graphique pour transformer des informations 3D en image 2D). L'utilisateur peut ensuite choisir la taille des points et la manière dont ceux-ci sont colorés (voir Figure 10)

La couleur des points affichés peut être calculée à partir de plusieurs éléments :

- position du point dans l'espace (distance à la caméra, altitude, etc.)
- choix d'un attribut (numéro d'écho Lidar, classification, etc.)

Le choix de la couleur en fonction de la valeur se fait grâce à un outil de création de cartes d'association de couleur (LUT : LookUp Table) qui associe à la valeur d'entrée, une couleur tirée dans un nuancier. zForest permet de réaliser des LUTs de manière interactive et de les sauvegarder au format des LUTs utilisé par le logiciel QGIS (voir Figure 11). Il permet également de charger les LUTs réalisées dans QGIS (format QML).

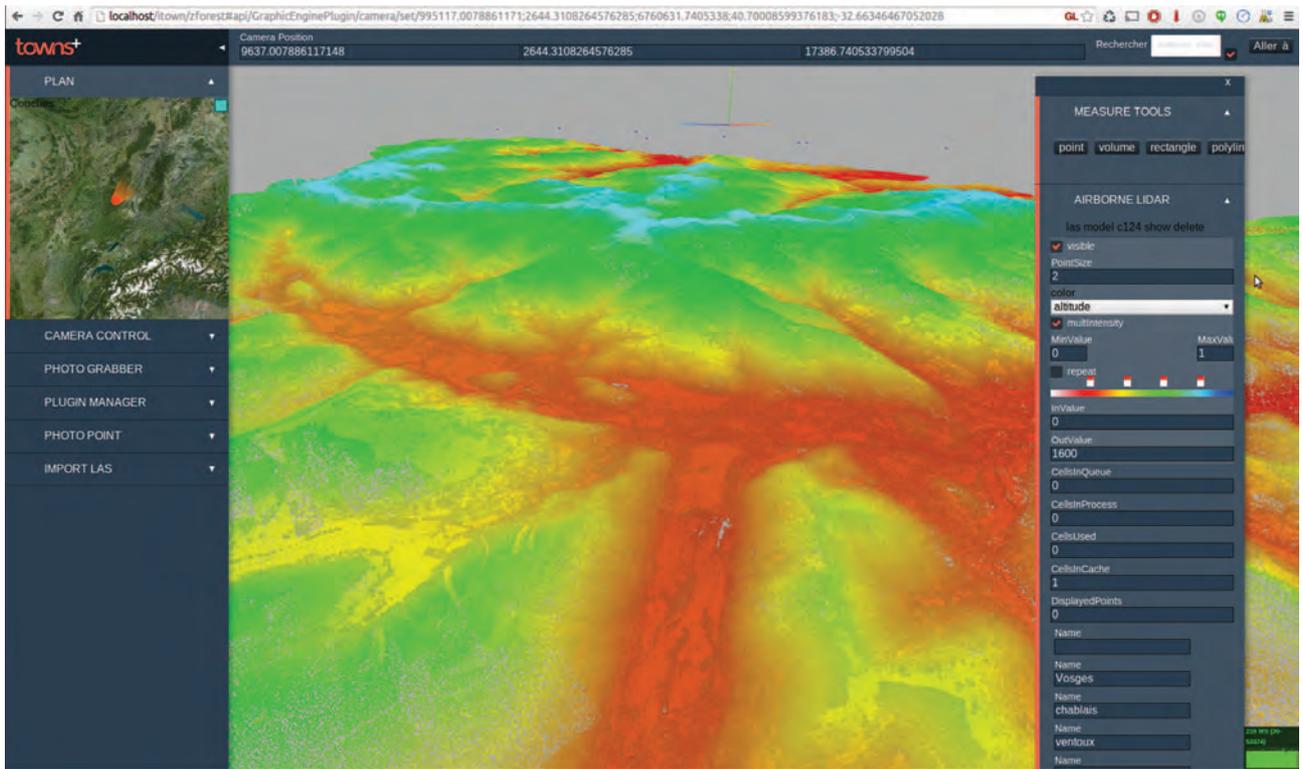


FIGURE 10: Vue de l'interface de paramétrage de l'affichage des points dans iTowns / zForest (à gauche). Ce menu permet de contrôler la taille des points affichés, l'attribut du point affiché (intensité, numéro d'écho, altitude. . .), mais également la couleur utilisée pour l'affichage.

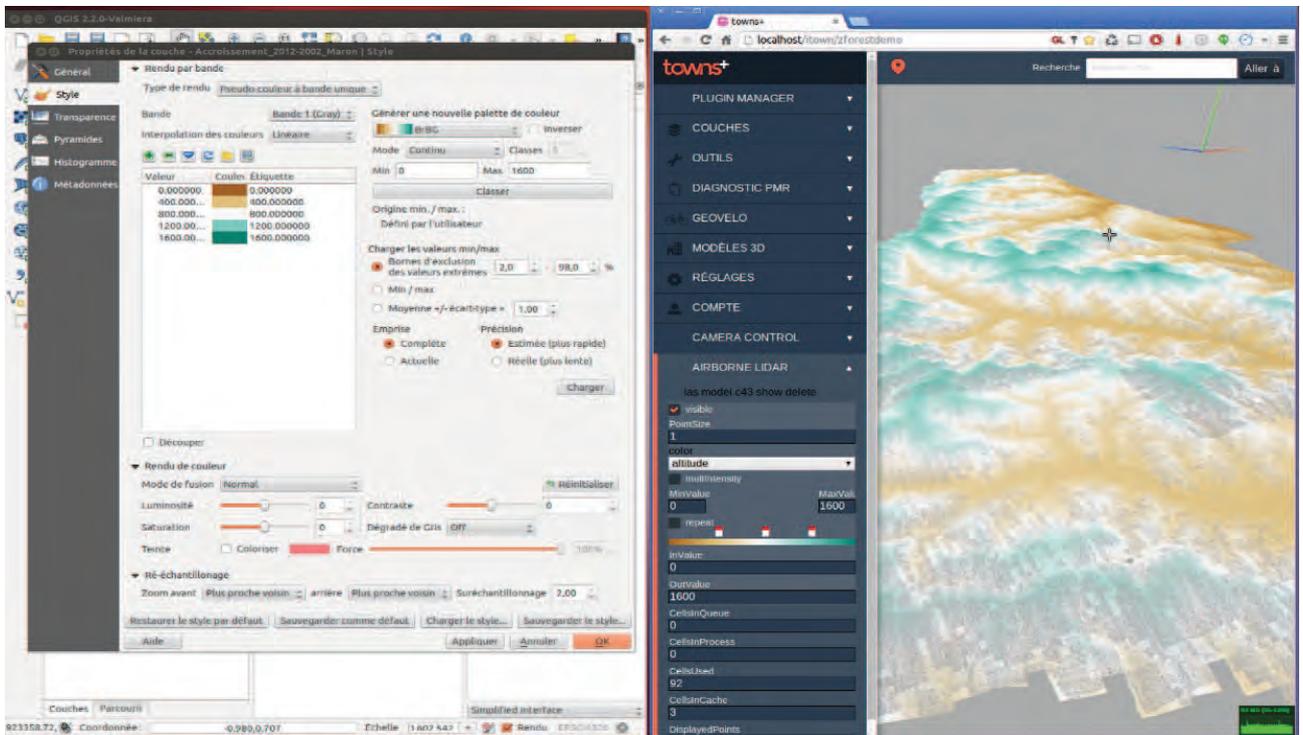


FIGURE 11: Import/Export de fichier de coloration avec QGis. À gauche, l'interface de paramétrage de table de valeurs de QGIS, qui permet d'associer une plage de couleur à une valeur d'entrée continue. À droite, une interface équivalente dans zForest permettant de paramétrer la gamme de couleurs ainsi que de choisir quel attribut est utilisée comme valeur d'entrée (ici, l'altitude des points Lidar). Le format de fichier des correspondances de couleur (aussi appelées LUT - LookUp Table) est commun aux deux outils.

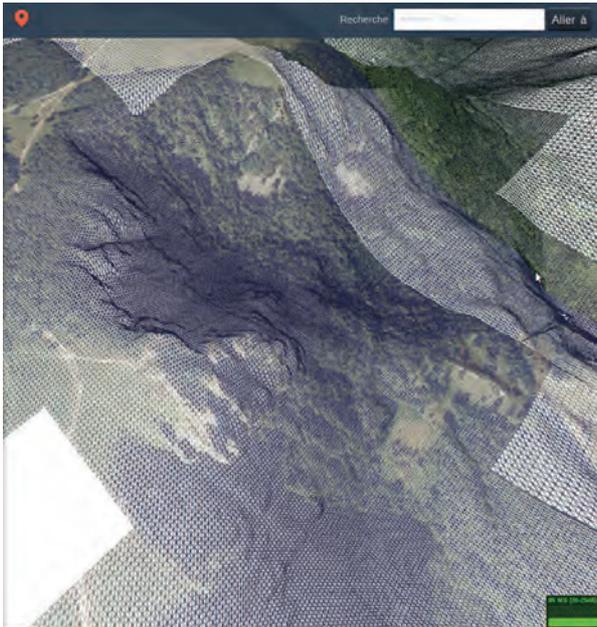


FIGURE 12: Affichage combiné Lidar + MNT + couche raster. zForest permet de choisir le serveur et la couche utilisée pour l'élévation. Il en va de même pour l'affichage couleur. L'outil exploite nativement les couches fournies par le Géoportail de l'IGN, ainsi qu'un serveur Web Map Service (WMS) dédié aux données produites dans le cadre de l'ANR FORESEE.. Pour ajouter un nouveau serveur dans zForest, il suffit que celui-ci soit conforme au standard WMS.

5. Co-visualisation et interactions avec les données

Comme nous l'avons dit précédemment, le nuage de points est parfois difficile à interpréter seul. Le croisement avec d'autres sources d'informations peut donner des clés de lecture pour son interprétation, comme il peut apporter de nouvelles clés de lecture à ces autres sources d'information. Ainsi, zForest propose plusieurs autres outils permettant d'afficher des informations complémentaires superposées aux nuages de points.

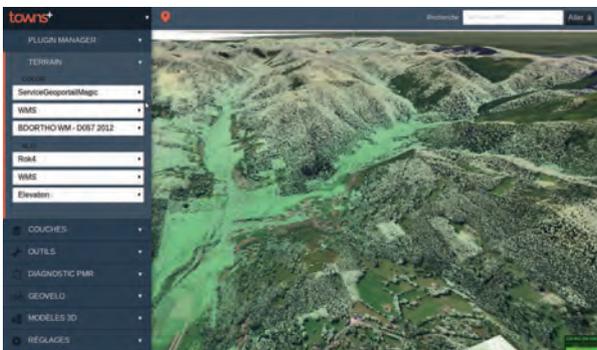


FIGURE 13: Nuage de points Lidar combiné au MNT et ortho-photo IGN.

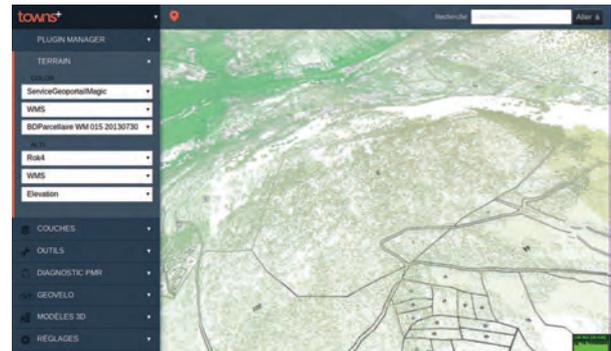


FIGURE 14: Nuage de points Lidar et cadastre.

5.1. Modèle Numérique de Terrain

Un modèle numérique de terrain fondé sur des grilles servies au standard Web Map Service (WMS) peut être affiché en même temps que le nuage de points. Le WMS est un protocole ayant la même finalité que le WMTS, mais avec un pouvoir d'expression plus grand : le serveur n'envoie pas les tuiles directement, mais peut effectuer des transformations (projection, changement de résolution...) sur celles-ci afin de produire une image spécifique. Ce MNT utilise les mêmes concepts que le système d'affichage de nuage de points, à savoir quadtree et gestion de requête (voir Figure 12).

L'utilisateur peut choisir la source de données d'élévation utilisée. Sur l'ANR FORESEE, nous avons le choix entre le MNT du Géoportail (2006) de l'IGN (au pas de 20 m) ou celui généré d'après le nuage de points Lidar (au pas de 1 m).

L'affichage d'un MNT en simultanément facilite la différenciation entre les points du sol et des arbres. Il permet également la mesure de la hauteur des arbres et constitue un support d'affichage pour des données raster ou vectorielles à proximité immédiates du nuage de point. Cet élément est d'autant plus important en montagne.

Le choix de la source de MNT est également important, car il permet de comparer la qualité et la résolution du MNT polygonal avec les échos Lidar enregistrés comme étant au sol. D'un point de vue plus opérationnel, le MNT peut être utilisé pour étudier des voies de passage potentielles d'engins forestiers (présence de talus, de rupture de pente, qui ne sont pas facilement visibles dans le nuage de point). L'utilisation combinée avec le Lidar donne une information complémentaire sur la densité végétale au-dessus du sol et donc l'accessibilité finale.

5.2. Drapage de données raster et vectorielle

Le MNT affiché peut être drapé par des sources d'information raster ou vectorielle rasterisée. Ainsi, toutes les couches du Géoportail sont disponibles, là encore, par le biais d'un service WMS. L'accès à d'autres serveurs est également possible dès lors qu'ils respectent le standard WMS, comme c'est le cas pour le serveur dédié à l'ANR FORESEE sur lequel certaines « cartes » résultant de travaux de recherche ont été déposées. Il est ainsi très intéressant de comparer le nuage de points

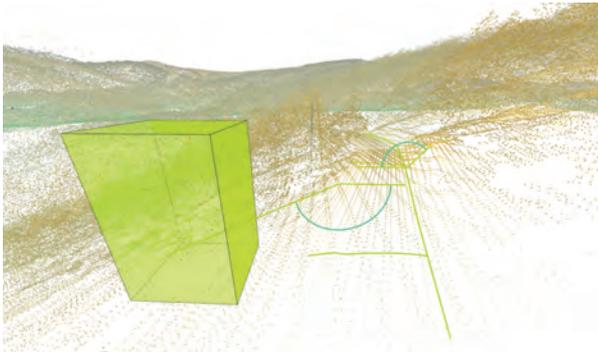


FIGURE 15: Primitives de dessin et mesure dans zForest. L'outil permet de dessiner des lignes et polygones et de mesurer leur longueur, mesurer des angles et des volumes simples (ici, la largeur d'une route, la boîte englobante d'un arbre. . .).

avec l'orthophoto ou les limites cadastrales (voir Figure 13 et 14). Dans le premier cas, la photo va aider à qualifier le nuage de point (type d'arbres, buissons, type de routes ou chemin). Dans le second, le cadastre va permettre d'avoir une idée relativement intuitive du volume de végétation sur une parcelle. L'affichage combiné d'autres types de données rasters ou vectorielles qui peuvent être le fruit de calculs statistiques s'appuyant sur le Lidar, permet également la comparaison de ces résultats avec la réalité "3D" du terrain. Ainsi dans le cadre de l'ANR FORESEE nous avons pu tester des cartes de surface terrières réalisées d'après Lidar et placettes de relevés dendrométriques, ainsi que des cartes chemins détectés en s'appuyant sur le MNT d'après Lidar.

5.3. Outils de dessin, mesure et profilage

La plateforme permet de dessiner des formes géométriques simples et de faire des mesures (voir Figure 15), par exemple la longueur d'un chemin. Ces outils de dessin s'appuient sur les coordonnées des points Lidar (ou à défaut, sur le MNT) pour être positionnés dans l'espace. À terme, cet outil pourrait être utilisé pour placer des annotations ou préparer un plan d'exploitation en s'appuyant sur la donnée 3D et sur un module d'extraction d'images (voir 5.4). Le MNT et le Lidar mettant en évidence les contraintes du terrain, il serait possible de dessiner des chemins d'accès potentiels, mesurer des courbures de routes ou marquer des aires de retournement pour les engins forestiers. La mise en place de primitives simples comme des cônes ou des cylindres permet d'isoler des portions de forêt voire même des arbres individuels, ou encore l'utilisation de gabarits grossiers d'engins pour planifier l'élagage d'une route sous couvert forestier.

En outre, les outils de dessin permettent de produire des extractions de profils de pente d'après le MNT, en suivant une courbe dessinée par l'utilisateur (voir Figure 16).

5.4. Outils de placement de photographies terrestres

zForest permet l'exploitation de photographies terrestres géolocalisées. L'outil utilise les données EXIF de



FIGURE 16: Extraction d'un profil de pente (utilisation du MNT IGN) le long d'un chemin dessiné dans le nuage de points.



FIGURE 17: Photographie terrestre placée dans zForest grâce aux données de géolocalisation EXIF.

la photographie pour placer celle-ci dans le nuage de points (voir Figure 17). L'utilisation de photographies terrestres apporte une lecture complémentaire du nuage de points et une réalité terrain appréciable pour son interprétation. L'outil proposé permet un positionnement simple de ces photographies. D'autre part, zForest permet la génération de captures d'écran géolocalisées. Ces images synthétiques incluent dans leurs EXIF les mêmes métadonnées qu'une photographie réelle. Ces images peuvent donc être intégrées dans des logiciels de gestion de banque d'images exploitant les informations de géolocalisation. Comme évoquées précédemment, ces images peuvent constituer une vue schématique intéressante pour un plan d'exploitation, et le fait qu'elles soient géolocalisées renforce leur traçabilité.

5.5. Service de données

L'utilisation de Node.js permet de s'appuyer côté serveur, sur les mêmes technologies que côté client, notamment le système d'évaluation du quadtree en fonction d'un volume englobant (frustum caméra, dans le cas d'une utilisation interactive). Cet aspect nous permet de mettre en place un service simple d'extraction de portions de nuage de points à la demande.

L'utilisateur peut sélectionner une zone dans zForest et un fichier Lidar sur mesure est construit en utilisant le quadtree côté serveur. Ce fichier est mis à disposition pour téléchargement. À terme, nous souhaitons que cet outil propose des services équivalents à ceux fournis

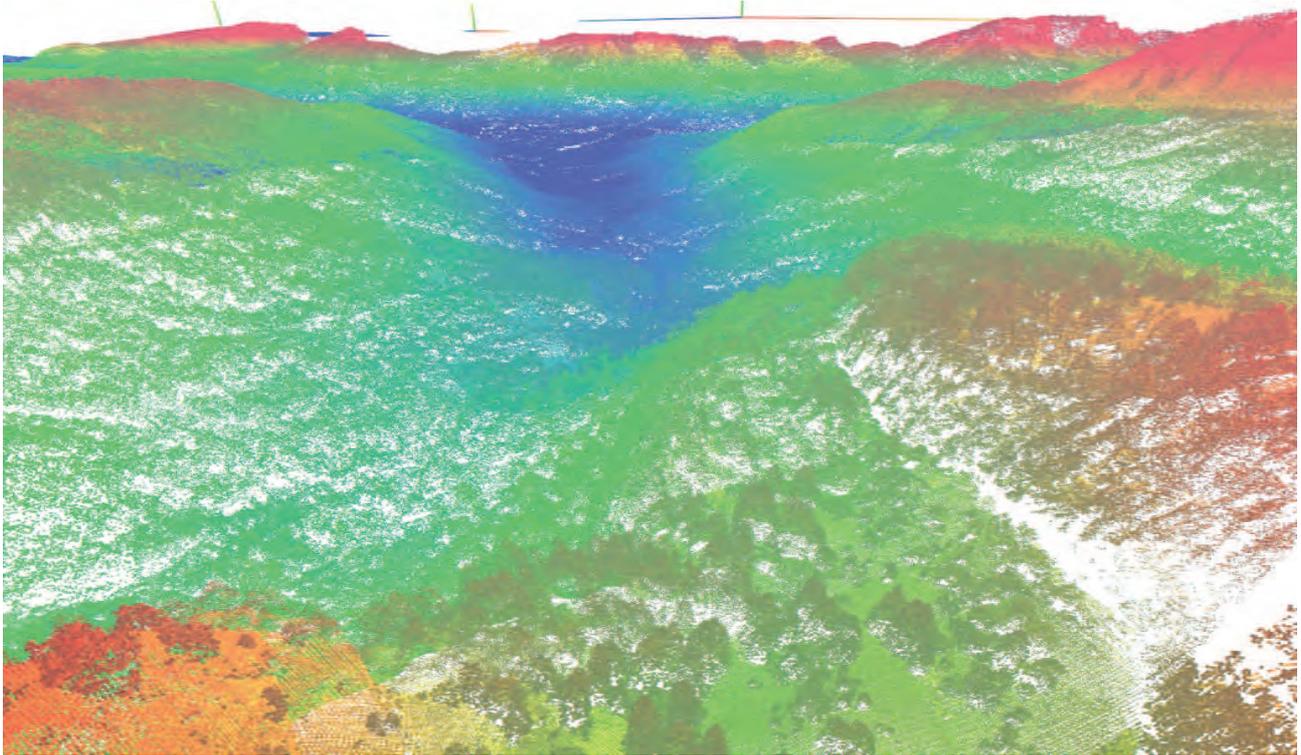


FIGURE 18: Vue du nuage de points d'une autre zone de test de l'ANR FORESEE : le massif des Bauges (iTowns / zForest). Taille totale : 1,3 milliard de points.

par la plateforme Earthexplorer (2010) de l'USGS (US Geological Survey - Organisme américain en charge de l'information géographique et sismique), à la différence que l'exploration et la sélection des données demandées dans zForest pourront se faire en 3D sur la donnée Lidar réelle, là où l'USGS ne propose qu'une vue 2D des zones disponibles en Lidar. Pour le chercheur, cet outil permet de naviguer "à l'oeil" dans la donnée, afin d'isoler puis "commander" les portions de nuage de point qui l'intéresse dans le cadre de sa recherche. Un tel outil permet également la réalisation de transects "Lidar".

6. Conclusion et travaux futurs

L'outil zForest permet d'explorer n'importe quel nuage de points aussi volumineux que celui produit lors de l'ANR FORESEE (disponible pour les Vosges et plusieurs autres zones présentant des natures de reliefs différents, par exemple le massif des Bauges présenté sur la Figure 18). L'application s'est montrée capable d'afficher de manière continue plusieurs acquisitions Lidar, dont une zone de 5,3 milliards de points, l'ensemble des acquisitions avoisinant les 10 milliards de points.

Il dispose d'une première version de système d'affichage du MNT, et permet de choisir n'importe quelle source raster comme donnée d'élévation ou comme donnée de couleur.

Il propose également des outils rudimentaires de mesure et d'extraction d'information.

zForest est un outil prototype qui montre (comme d'autres outils, notamment Potree) que l'exploration de

grands nuages de points Lidar est possible dans un navigateur Web. Le nombre d'applications similaires apparues en parallèle du développement de cet outil témoigne d'un engouement et d'un intérêt pour ces techniques de numérisation et ces méthodes d'accès « grand public » aux données Lidar ou nuages de points issus de photogrammétrie. La démocratisation des drones est également un élément stimulant en terme de numérisation 3D et les besoins en visualisation (et échange de données) en ligne vont grandissant. zForest a la capacité de rassembler en un seul endroit simple d'accès, des données d'origines diverses afin de permettre une interprétation combinée de la donnée Lidar à grande échelle. C'est ce qui en fait une plateforme unique en son genre aujourd'hui dans le paysage des outils de visualisation dans le navigateur internet. Du côté de l'IGN, le projet d'affinage à 1m du MNT français en utilisant le Lidar aéroporté va permettre la création d'une base de données Lidar très intéressante pour d'autres applications (et notamment la foresterie). Nous espérons que les travaux menés sur zForest constitueront un axe de réflexion pour la réalisation d'une plateforme type « Géoportail » d'exploration de cette nouvelle donnée.

Remerciements

Ce projet de développement a été financé par l'Agence Nationale de la Recherche (ANR) à travers le projet ANR FORESEE (ANR-2010-BIOE-008).

Références

- Cellier, F., 2014. Modélisation et calcul parallèle pour le Web SIG 3D. Thèse de doctorat en informatique, Université Lyon1.
URL <http://liris.cnrs.fr/publis/?id=6454>
- CloudCompare, 2006. 3d point cloud and mesh processing software open source project.
URL <http://www.danielgm.net/cc/>
- Devaux, A., Paparoditis, N., 2010. Increasing interactivity in street view web navigation systems. Dans : ACM MULTIMEDIA 2010. Firenze, Italy.
- Devaux, A., Paparoditis, N., Brédif, M., 2012. A web-based 3d mapping application using WebGL allowing interaction with images, point clouds and models. Dans : 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2012). Redondo Beach, CA, USA.
- Dick, C., Krüger, J., Westermann, R., 2009. Gpu ray-casting for scalable terrain rendering. Dans : Proceedings of EUROGRAPHICS. Vol. 50.
- Earthexplorer, 2010. Provides search and order capabilities for satellite images, aerial photographs, and cartographic products.
URL <http://earthexplorer.usgs.gov/>
- Elseberg, J., Borrmann, D., Nüchter, A., 2013. One billion points in the cloud—an octree for efficient processing of 3d laser scans. ISPRS Journal of Photogrammetry and Remote Sensing 76, 76–88.
- Ferraz, A., Mallet, C., Chehata, N., 2015. Détection de la déserte forestière en milieu montagneux par lidar aéroporté (soumis). Revue Française de Photogrammétrie et de Télédétection (submitted).
- FugroViewer, 2011. A 3d geodata viewer.
URL <http://www.fugroviewer.com/>
- Géoportail, 2006. Le portail des territoires et des citoyens.
URL <http://www.geoportail.gouv.fr/accueil>
- LasTools, 2006. Efficient tools for lidar processing.
URL <http://www.cs.unc.edu/~isenburg/lastools/>
- Node.js, 2009. A platform built on chrome's javascript runtime for easily building fast, scalable network applications.
URL <http://nodejs.org/>
- Paparoditis, N., Papelard, J.-P., Cannelle, B., Devaux, A., Soheilian, B., David, N., 2012. Stereopolis ii : A multi-purpose and multi-sensor 3d mobile mapping system for street visualisation and 3d metrology. Revue Française de Photogrammétrie et de Télédétection 200 : 69-79.
- Potree, 2012. WebGL pointcloud renderer.
URL <http://potree.org/>
- Scheiblauer, C., 2008. Domitilla catacomb walkthrough - dealing with more than 1 billion points. Dans : Proceedings of the 13th International Congress on Cultural Heritage and New Technologies. Tagungsreihe "Kulturelles Erbe und Neue Technologien" - Workshop "Archäologie und Computer". Cultural Heritage & New Technologies.
URL <http://www.cg.tuwien.ac.at/research/publications/2008/Scheiblauer-2008-DCW/>
- Terrascan, 2002. Software for lidar data processing and 3d vector data creation.
URL <http://www.3dlasermapping.com/software/terrascan>
- Three.js, 2010. A javascript 3d library which makes WebGL simpler.
URL <http://threejs.org/>
- Wiendl, S., Dick, S., 2013. Gpu-aware hybrid terrain rendering. Dept. Comput. Sci., Univ. Munich, Germany, Tech. Report.