

# TVPD : un jeu de données pour évaluer les algorithmes d'estimation de points de fuite (The Toulouse Vanishing Points Dataset)

Vincent Angladon<sup>1,2</sup>

Simone Gasparini<sup>1</sup>

Vincent Charvillat<sup>1</sup>

<sup>1</sup> Université de Toulouse ; INPT – IRIT

<sup>2</sup> Telequid

118 Route de Narbonne, F-31062 Toulouse, France

vincent.angladon@irit.fr

## Résumé

Nous présentons à travers cet article un nouveau jeu de données comprenant un corpus de photos de scènes dites de Manhattan prises avec un iPad Air 1. Ce jeu de données permet d'évaluer des algorithmes d'estimation de points de fuite. À la différence des jeux de données existants, des données inertielles issues des différents capteurs de l'iPad, synchronisées avec la caméra sont également présentes. Ces données inertielles peuvent être utilisées pour améliorer les performances d'un algorithme d'estimation de points de fuite. Enfin, nous présentons un nouvel algorithme d'estimation de régions d'incertitude sur les points de fuite estimés à partir des segments vérités-terrain. Le Toulouse Vanishing Points Dataset est accessible sur la page <http://ubee.enseeiht.fr/tvdp>

## Mots Clef

Jeu de données, point de fuite, capteurs inertiels, smartphone.

## Abstract

In this paper we present the Toulouse Vanishing Points Dataset, a public photographs database of Manhattan scenes taken with an iPad Air 1. The purpose of this dataset is the evaluation of vanishing points estimation algorithms. Its originality is the addition of Inertial Measurement Unit (IMU) data synchronized with the camera under the form of rotation matrices. Moreover, contrary to existing works which provide vanishing points of reference in the form of single points, we computed uncertainty regions. The Toulouse Vanishing Points Dataset is publicly available at <http://ubee.enseeiht.fr/tvdp>

## Keywords

Dataset, vanishing point, IMU sensor, mobile device.

## 1 Introduction

L'interprétation d'une image nécessite l'analyse de la géométrie de sa scène et recouvre divers sujets tels que la robotique domestique, l'analyse sémantique de scènes urbaines

et l'étude de l'agencement spatial de scènes intérieures. C'est un problème difficile dû à l'absence d'information de profondeur dans l'image. En effet, la projection perspective est une transformation non inversible entre l'espace tridimensionnel de la scène et le plan image bidimensionnel. L'analyse des éléments géométriques d'une image joue ainsi un rôle important dans l'étude de l'agencement spatial de la scène.

Les *points de fuite* sont des objets géométriques qui peuvent donner des indications précieuses pour interpréter une image. Les droites parallèles de la scène sont transformées via une projection projective en un faisceau de droites qui s'intersectent en un point nommé point de fuite. Ce point image est la projection de l'intersection à l'infini de ces droites parallèles. Lorsque la caméra est calibrée, l'image d'un point de fuite donne la direction 3D de ce faisceau de droites dans la scène. La plupart des scènes réalisées par l'homme contiennent trois directions dominantes orthogonales. Elles sont appelées « Scènes de Manhattan » [12] et sont donc constituées de trois ensembles de droites parallèles entre elles.

La détection des trois points de fuite associés à ces droites permet de calibrer la caméra [9, 37] et d'obtenir son orientation dans la scène [2, 21, 24]. Plus récemment, les points de fuite font partie intégrante de nombreuses approches visant à interpréter des images de scènes intérieures et extérieures [23, 27, 20] et à estimer l'agencement spatial d'une pièce (détermination du sol et des murs) [29, 32]. Ce dernier sujet nous intéresse plus particulièrement car il rejoint notre objectif final : *la reconstruction 3D de scènes intérieures sur dispositifs mobiles*. Ce sujet complexe a été découpé en plusieurs sous-problèmes dont la localisation de l'utilisateur, la segmentation des murs et l'affichage des résultats par l'augmentation de la scène affichée. La résolution de ces trois problématiques passe par l'extraction d'informations géométriques, et en particulier les points de fuite.

L'utilisation d'un smartphone ou d'une tablette permet d'envisager l'exploitation de données inertielles fournies le plus souvent par la combinaison d'un accéléromètre, d'un

gyroscope et d'un magnétomètre. Leur utilisation permet d'obtenir en temps réel l'orientation du téléphone et d'améliorer l'estimation du déplacement de la caméra qui ne peut être toujours fiable. En effet, les images produites par les caméras CMOS des smartphones peuvent comporter des flous de bouger et des déformations dues au phénomène de l'obturateur déroulant (rolling shutter). De plus, le manque d'éclairage et l'absence de texture ne permet pas d'extraire des caractéristiques visuelles dans l'image, rendant le problème d'estimation de la pose de la caméra très difficile. L'utilisation conjointe de ces capteurs avec de la vision par ordinateur commence à se développer aujourd'hui, notamment dans les applications utilisant de la réalité augmentée ou effectuant des reconstructions 3D [34].

Les données visuelles et inertielles sont complémentaires : ces dernières peuvent certes être obtenues en temps réel mais elles sont bruitées et soumises à divers biais, tandis que les données visuelles sont plus stables et précises mais nécessitent des temps de calculs plus importants. La donnée du vecteur gravité calculé à partir des capteurs inertiels peut être utilisée pour guider et simplifier la détection des points de fuite.

Dans cet article, nous présentons une nouvelle banque d'images de scènes intérieures et extérieures dites de « Manhattan » prises avec un iPad Air 1 donc équipé de capteurs inertiels. Les données issues de ces capteurs sont enregistrées dans les images elles-mêmes sous la forme d'une matrice de rotation. Des vérités-terrain pour les points de fuite de chaque image sont générées, mais au lieu d'être présentées sous la forme de points pour chacune des trois directions orthogonales, nous proposons des régions d'incertitude. Ces régions calculées à partir des segments vérités-terrain sont des polygones qui contiennent les trois points de fuite orthogonaux solution exacte obtenus si les segments avaient été parfaitement annotés.

Nous proposons deux contributions : une nouvelle méthode de calcul de points de fuite sous forme de régions d'incertitude, et la création d'un jeu de données permettant d'évaluer des algorithmes d'estimation de points de fuite. À notre connaissance, il n'existe pas d'autre banque d'images contenant des données inertielles.

Ce papier est organisé de la façon suivante : La section 2 présente un aperçu des travaux sur les points de fuite et montre comment utiliser des données a priori provenant de capteurs inertiels. La section 3 décrit et critique les jeux de données existants utilisés pour l'évaluation d'algorithmes d'estimation de points de fuite. La section 4 explique comment nous calculons nos régions d'incertitude à l'aide des vérités-terrain tandis que la section 5 décrit la méthodologie utilisée pour l'acquisition des données.

## 2 Travaux antérieurs

Dans les sections suivantes, nous résumons brièvement les travaux antérieurs sur l'estimation de points de fuite. Nous présentons ensuite les données fournies par les capteurs inertiels et comment elles peuvent être utilisées pour sim-

plifier la détection de points de fuite.

### 2.1 Estimation de points de fuite

La détection de points de fuite repose sur l'extraction et l'analyse de caractéristiques visuelles telles que des gradients d'intensité lumineuse, des contours, des droites ou des segments qui peuvent être regroupés en grappes. Chaque grappe contient un faisceau de droites 2D de l'image correspondant à des droites 3D parallèles de la scène.

Le regroupement en grappes des segments permet de définir le nombre de points de fuite de la scène et leur position via un processus de minimisation détaillé par la suite. Inversement, les points de fuite permettent de déterminer le regroupement en grappes des segments. Les segments d'une grappe sont ceux situés à une distance raisonnable du point de fuite considéré. Cette distance segment-point est définie à partir d'une fonction appelée *mesure de consistance*. La difficulté consiste à effectuer ce regroupement en grappes des segments sans connaître au préalable les points de fuite.

L'utilisation de données a priori sur la scène permet de simplifier ce problème. Par exemple, la connaissance des paramètres intrinsèques de la caméra permet de transposer la notion d'orthogonalité de la scène dans l'image, ce qui ajoute des contraintes sur la solution. Dans cet article nous prenons seulement en considération les scènes dites de Manhattan, ce qui réduit le problème à la recherche d'un triplet de trois points correspondant à des directions orthogonales dans la scène. Enfin, la connaissance de l'orientation partielle de la caméra donnée par des capteurs inertiels permet de réduire l'espace de recherche comme nous le verrons dans la prochaine section.

Généralement, la plupart des méthodes d'estimation de points de fuite proposées dans la littérature utilisent des segments comme caractéristiques visuelles qui peuvent être extraits via des algorithmes de traitement d'images avancé tel que LSD [18] qui est basé sur une approche à contrario. D'autres auteurs comme Choi [10] utilisent des gradients d'intensité lumineuse comme caractéristique visuelle qui indiquent l'orientation locale sur une zone donnée et permettent d'estimer un seul point de fuite dominant.

L'estimation de points de fuite se décompose généralement en deux étapes : le *regroupement en grappes* des segments et l'*estimation des points de fuite* pour chaque grappe. Diverses techniques de regroupement ont été suggérées dans la littérature : transformée de Hough [5], RANSAC [30, 38, 1], et J-Linkage [35]. La deuxième étape définit le point de fuite de chaque grappe comme le point qui minimise une fonction d'erreur  $W$  pour chaque segment d'une grappe. Plusieurs fonctions  $W$  ont été proposées : somme de distances point-segment [3, 30, 35], somme de fonctions d'erreur d'orientation [14, 28, 31], ou fonction d'erreur probabiliste [11, 39]. Le triplet final est alors choisi parmi les solutions possibles, ou comme dans [30] la contrainte d'orthogonalité peut être exploitée au cours de la procé-

de l'estimation des points de fuite.

Certaines méthodes ne suivent pas un schéma en deux temps et essaient plutôt de résoudre le problème globalement : [6] essaie de trouver la rotation (*i.e.* le triplet de points de fuite) qui maximise le nombre de segments clustérisés. Antunes [3] suit une approche globale dans laquelle le regroupement et l'estimation des points de fuite sont estimés en même temps via la résolution d'un problème d'optimisation de chaînes logistiques (Uncapacitated Facility Location). Une approche globale courante est l'utilisation d'un schéma espérance-maximisation comme proposée par Kosecka [24].

## 2.2 Données inertielles

Au cours des dernières décennies, les progrès en micro-électronique ont conduit à l'emploi massif de capteurs miniatures devenus bon marché dans les smartphones et les tablettes. Les capteurs inertiels sont constitués en général d'un accéléromètre mesurant l'accélération de l'appareil, de gyroscopes mesurant le moment cinétique et d'un magnétomètre mesurant son orientation relativement au pôle Nord magnétique. Les données issues de ces capteurs sont généralement précises à court terme. En contrepartie, les données sont soumises à divers biais : facteur d'échelle, bruit blanc, dépendance à la température qui sont difficiles à caractériser [16]. De plus, les données du gyroscope doivent être intégrées pour obtenir des mesures d'angle, ce qui conduit à une dérive des valeurs calculées. La redondance d'information fournie par tous ces capteurs permet d'utiliser des techniques de filtrage tels que des filtres de Kalman [22] afin d'atténuer les dérives et le bruit des données obtenues.

Dans cet article nous prenons en considération les données de ces capteurs permettant de calculer l'attitude de l'appareil, ou dit autrement, la matrice de rotation de la caméra respectivement à un référentiel terrestre. Le kit de développement (sdk) des systèmes d'exploitation permettent généralement d'obtenir cette matrice de rotation  $R$  après fusion des données des capteurs inertiels. À partir de cette matrice, il est aisé de calculer le point de fuite vertical (*i.e.* le zénith) :  $\mathbf{v}_{\text{zenith}} = K R [0 \ 0 \ 1]^T$ , où  $K$  est la matrice des paramètres intrinsèques de la caméra.

Le point de fuite vertical est dual à la ligne d'horizon qui est la projection du plan passant par le centre de la caméra et de normale parallèle à la direction du point de fuite. Comme les deux autres points de fuite de la scène de Manhattan sont orthogonaux au zénith, l'estimation de leur position peut être réduite à la recherche d'un point de fuite le long de la ligne d'horizon.

## 3 Jeux de données existants

Il existe à ce jour deux jeux de données notables utilisés pour la comparaison d'algorithmes de points de fuite. Le *York Urban Database* [14, 15] publié en 2008 a été utilisé dans de nombreux travaux pour évaluer les algorithmes proposés. Il contient 102 images de scènes inté-

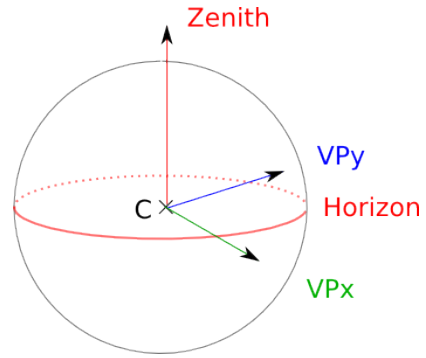


FIGURE 1 – Les trois points de fuites forment un repère orthogonal.  $C$  est le centre optique de la caméra qui est calibrée. La connaissance du zénith permet de limiter la recherche des deux points de fuite orthogonal au zénith ( $VP_x$  et  $VP_y$ ) sur la ligne d'horizon.

rieures et extérieures dites de Manhattan ainsi que des segments créés à la main et regroupés selon les trois directions de Manhattan. Les points de fuite sont estimés à partir de ces segments en utilisant l'algorithme proposé par [11]. Cette méthode statistique estime les points de fuite en prenant en considération les différentes grappes de façon séparée. Un triplet de vecteurs orthogonaux est ensuite estimé par une méthode de moindres carrés pondérés sur les directions de Manhattan précédemment estimées. Ainsi la solution obtenue n'est pas nécessairement optimale au sens de la distribution statistique estimée sur les intersections des droites correspondant aux segments annotés. Il arrive ainsi que la solution donnée soit assez éloignée du lieu des intersections de ces segments, comme on peut le constater sur la Figure 2. Dans ce cas précis, les directions de Manhattan fournies sont erronées et ne peuvent être envisagées comme référence pour évaluer et comparer des algorithmes de détection de points de fuite.

Plus récemment, le *PKU Campus Database* [25, 26] contenant 200 photographies de scènes de Manhattan a été proposé. Ce jeu de données est inspiré du *York Urban Database* dans le sens où il suit les mêmes conventions de nommage, les images ont la même résolution et les points de fuite de référence sont calculés de la même façon et souffrent donc du problème d'orthogonalisation postérieure mentionné précédemment.

Dans un souci d'exhaustivité, nous mentionnons également le *Eurasian Cities Dataset* [4, 36] qui contient 103 images de scènes extérieures. Cependant cette banque d'images a été construite pour un objectif sensiblement différent : l'étude de scènes dites d'Atlanta.

Les algorithmes d'estimation de points de fuite sont généralement évalués en comparant la position des points estimés avec des points de fuite de référence fournis par un jeu de données. Cette méthode suppose que les points de fuite de référence sont exacts, ce qui n'est pas le cas des jeux de données mentionnés précédemment, et généralement difficile à prouver. De façon générale, il est impossible d'être



FIGURE 2 – Cette figure montre les points de fuites et les lignes d’horizons fournis par le York Urban Database, avant et après orthogonalisation. Un des points de fuite estimé par [11] est représenté par un carré vert. La ligne d’horizon en rouge est calculée par la même méthode. Le point de fuite vert en forme de losange et la ligne d’horizon en pointillé rose sont obtenus après orthogonalisation de la solution précédente. On peut observer que ce dernier est situé relativement loin du lieu des intersections des différentes droites (en vert) de la scène.

exact. Comme le fait constater Xu [39], même avec des segments créés de façon très précise, il y a beaucoup d’incertitudes et d’inconnues qui rentrent en jeu et affectent l’estimation des points de fuite : les imperfections du système optique, les erreurs de calibrage de l’appareil photo, l’extraction des segments, *etc.* Seules les images de synthèses (où tous les paramètres sont connus) permettent de calculer des vérités-terrain *exactes* pour les points de fuite. Nous proposons une approche différente qui consiste à calculer des régions d’incertitude pour les points de fuite au lieu de points. Cette région peut être utilisée pour accepter ou rejeter la solution d’un algorithme estimant des points de fuite (le point est respectivement à l’intérieur ou à l’extérieur de la région).

#### 4 Création des vérités terrain

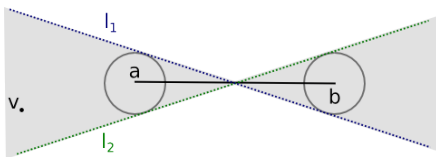


FIGURE 3 – L’incertitude d’un segment dessiné à la main est modélisée par deux régions circulaires centrées sur les extrémités du segment a et b. Les droites qui relient toutes les extrémités possibles contenues dans ces disques engendrent une région appelée *double wedge* (représentée en gris). Cette région est délimitée par deux droites  $l_1$  et  $l_2$  et contient le point de fuite  $v$  associé à ce segment.

L’incertitude d’un segment vérité terrain provient de ses extrémités et peut être modélisée par des régions circulaires d’incertitude centrées sur ses extrémités (Figure 3). Shufelt [33] fut le premier à introduire une modélisation de cette incertitude pour calculer des points de fuite. Il considère

que la position réelle de l’extrémité d’un segment est un point de coordonnées réelles situé dans son pixel (qui est de coordonnées entières). Les droites reliant les extrémités possibles situées dans leur pixel définissent un *double wedge* [7] (la région grisée dans la Figure 3 qui correspond à un cône du plan). Ce double wedge contient le segment *exact* qui aurait du être annoté et contient donc une droite passant par le point de fuite recherché. Dans notre cas, nous supposons que tous nos segments annotés à la main ne sont pas aberrants, ainsi l’intersection de tous les double wedges associés à une direction de Manhattan donnée définit une région dans laquelle est situé le point de fuite recherché (Figure 4). Pour calculer les points de fuite, Shufelt utilise la sphère de Gauss [5] qui est découpée en accumulateurs sur laquelle il projette les double wedges en incrémentant les accumulateurs correspondants. Les maxima de cet espace de vote sphérique représentent les directions de Manhattan.

Plus récemment, Xu [39] a introduit une mesure de consistance probabiliste qui modélise avec une gaussienne 1D l’incertitude sur la position des extrémités des segments. Cette distance est ensuite utilisée dans un schéma espérance-maximisation pour déterminer les points de fuite. Contrairement à Xu, notre approche est géométrique car notre objectif est de calculer des régions d’incertitude pour les points de fuite, plutôt que des points. Lors de la création d’un segment, nous ne supposons pas qu’il est moins probable de commettre une erreur de 2 pixels que de 1 pixel sur la position de ses extrémités. C’est une des raisons pour laquelle, dans nos régions d’incertitude, au lieu de mettre en avant un point particulier, nous considérons que tous les points de fuite du polygone sont équiprobables. À la différence de [33], nous calculons l’intersection exacte des double wedges plutôt que d’utiliser des accumulateurs.

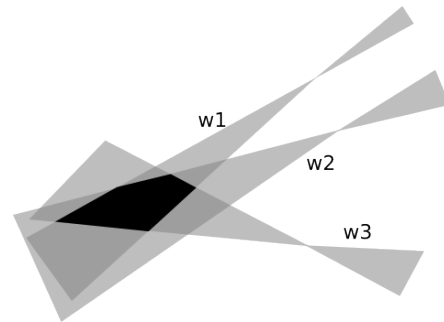


FIGURE 4 – L’intersection des double wedges  $w_1$ ,  $w_2$ ,  $w_3$  associés à la même direction de Manhattan définit un polygone convexe dans lequel est situé le point de fuite recherché.

L’utilisation de double wedges pour modéliser les incertitudes sur les segments permet de prendre en compte *facto* la longueur de ces derniers. Généralement, les segments longs sont davantage intéressants car les erreurs sur leurs extrémités perturbent moins la solution. En effet, ces segments ont un double wedge plus fin qui contribue à ré-

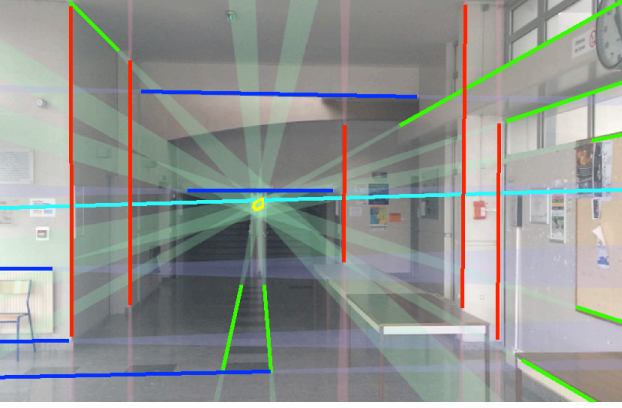


FIGURE 5 – Le polygone jaune est l’intersection des doubles wedges verts. L’horizon calculé à partir des données IMU est représenté en cyan.

duire la région d’incertitude associée au point de fuite. Au contraire, les segments courts ont un double wedge plus évasé qui ne participe pas à la réduction de la solution. Pour calculer l’intersection des double wedges, nous reformulons ce problème en terme d’opérations booléennes sur des demi-plans. Soient  $I_1$  et  $I_2$  les droites délimitant le double wedge associé au segment  $[a\ b]$  (voir Figure 3). Sans perte de généralité, considérons les demi-plans  $h_1$  et  $h_2$  de frontières respectives  $I_1$  et  $I_2$  et contenant chacun le point  $a$  soit encore  $\forall i \in \{1, 2\}, h_i \stackrel{\mathcal{D}}{=} \{p \in \mathbb{R}^3 | l_i^T p \leq 0\}$  et  $\bar{h}_i \stackrel{\mathcal{D}}{=} \{p \in \mathbb{R}^3 | l_i^T p > 0\}$  où les points  $p$  sont en coordonnées homogènes. Le double wedge  $w$  associé à  $[a\ b]$  est défini par

$$w = (h_1 \cap h_2) \cup (\bar{h}_1 \cap \bar{h}_2),$$

où  $\bar{h}_i$  représente le complémentaire de  $h_i$ . L’intersection des double wedges de tous les segments peut être calculée par intersection et union des demi-plans  $h_i$  de chaque segment, ce qui est un problème bien connu en géométrie algorithmique traité dans [7]. Le calcul de l’intersection est effectué dans le plan projectif, ce qui revient à travailler dans la sphère de Gauss, nous permettant de gérer les intersections de droites parallèles dans l’image et donc les points de fuite à l’infini.

## 5 Acquisition des données

Le jeu de données présenté contient à ce jour 114 photographies constitué de 40 scènes intérieures et 74 scènes urbaines de la région toulousaine. Les photos ont été prises à différents moments de la journée, par conséquent avec des expositions différentes. La Figure 6 montre quelques photos de notre banque d’images avec les différentes données que nous fournissons : région d’incertitude des points de fuite et ligne d’horizon calculée à partir des données inertielles.

Nous avons pris les différentes photos avec un iPad Air 1 sous iOS 8 tenu en mode paysage, en utilisant une résolution



FIGURE 6 – Quelques photos du jeu de données avec leurs segments annotés à la main (en rouge, vert, bleu). En cyan : la ligne d’horizon calculée à partir des données inertielles. Les polygones des régions d’incertitude des points de fuite associés aux segments verts sont représentés en jaune. Sur la première ligne, les lignes d’horizon n’intersectent pas les polygones jaunes à cause du biais des données inertielles.

de 1920x1080 pixels et les paramètres iOS de capture suivants : balance des blancs automatique, exposition automatique et autofocus désactivé. L’auto-focus a été désactivé car il peut introduire un décalage aléatoire entre le moment où l’on appuie sur le déclencheur et le moment effectif où la photo est prise.

Au lieu d’utiliser les données brutes des accéléromètres, gyroscopes et du magnétomètre, nous avons utilisé la classe `CMDeviceMotion` du SDK iOS qui fournit des méthodes de haut niveau calculant le vecteur gravité et l’orientation de l’appareil via des algorithmes de fusion de données capteur non précisés dans la documentation officielle. Un échantillonnage de 30Hz a été choisi pour l’acquisition des données inertielles. Nous avons développé une application spécifique pour enregistrer l’orientation de la tablette lors de la prise des photos, son code source est disponible sur le site de notre jeu de données.

**Calibrage de la caméra et des capteurs inertiels** Les paramètres intrinsèques de la caméra ont été estimés en utilisant la toolbox Matlab de Bouquet [8]. Les distorsions optiques mesurées étant très faibles, nous avons choisi de ne pas en tenir compte. Nos expériences avec les capteurs inertiels sur l’iPad nous ont révélé que l’on pouvait obtenir dans le pire cas 2 degrés d’erreur sur les valeurs du roulis et du tangage. Le biais de ces capteurs est visible sur la Figure 6 où les lignes d’horizon calculées à partir de ces données n’intersectent pas les polygones des régions d’incertitude des points de fuite associés aux directions de Manhattan représentées en vert. En outre, aucun étalonnage de ces capteurs n’a été effectué car nos observations ont révélé que les défauts de planéité du sol engendrent des er-



reurs plus importantes que la précision de ces capteurs.

Un problème connu affectant les appareils mobiles est la synchronisation des données inertielles avec les images fournies par la caméra. Pour des raisons économiques et techniques, les données issues des capteurs des dispositifs mobiles ne sont pas horodatées avec une horloge commune. Ham [19] montre que les conséquences de cette absence de synchronisation peut être désastreuse dans la cas de la recherche du facteur d'échelle d'un objet par fusion de données inertielles avec des images de la caméra. Nous avons mesuré un retard moyen des données inertielles par rapport aux images de la caméra de 16ms avec un écart-type de 140ms dans le cas où la charge système est élevée, contre 5ms et un écart type de 18ms dans le cas contraire. Lorsque la charge système est faible, on peut constater qualitativement que les données inertielles sont en retard ponctuellement d'une frame sur les données visuelles. Afin de tenir compte de cette incertitude temporelle et de lisser les données, nous calculons la matrice d'orientation moyenne de l'iPad via la méthode de Curtis [13] sur une fenêtre temporelle couvrant le retard moyen que nous avons mesuré. Il existe des méthodes de calibrage temporel plus précises comme [17]. Cette méthode n'est pas adaptée à notre cas puisque nous utilisons des données inertielles déjà fusionnées et notre acquisitions vidéo à 30Hz (limitation matérielle) est visiblement trop faible pour obtenir un calibrage correct.

**Création des segments** Pour dessiner précisément les segments vérités-terrain, nous avons conçu une application web (Figure 7) similaire à celle développée par Denis [14]. Nous supposons que nous commettons dans le pire des cas une erreur de 4 pixels sur la position des extrémités de ces segments. Cette valeur a été déterminée expérimentalement sur notre jeu de données, de sorte que pour toutes les photos, l'intersection des double wedges ne soit pas vide et contienne les points de fuite calculés à partir de la mesure de consistance définie par Antunes [3]. Nous avons également effectué des comparaisons avec la banque d'images du York Urban Database (Figure 8). Sans surprise, les points de fuite calculés par la méthode de Collins et Weiss [11] sont généralement situés dans nos polygones, alors que ce n'est pas nécessairement le cas pour les points de fuite orthogonalisés. En effet, le processus d'orthogonalisation employé ne prend pas en compte les segments (*c.f.* section 3).

Le code source de cette application est disponible sur la page web de notre banque d'images.

## 6 Conclusion

À travers cet article, nous avons présenté une nouvelle banque de photographies de scènes dites de Manhattan permettant l'évaluation et la comparaison d'algorithmes d'estimation de points de fuite. Ce jeu de données est le seul à notre connaissance à inclure des données inertielles. Au lieu de fournir des vérités-terrain *exactes* pour les points de fuite sous formes de points, ce qui est délicat compte

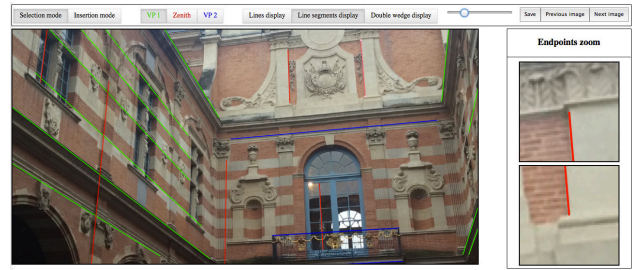


FIGURE 7 – L'application web permettant d'annoter les images avec des segments.

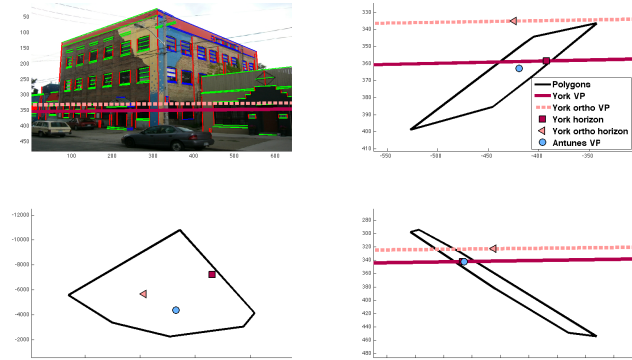


FIGURE 8 – Comparaison avec le York Urban Database. La ligne d'horizon en rouge et les points de fuite (carrés rouges) calculés avec [11], les points de fuite calculés avec [3] (cercles bleus) sont situés dans nos régions d'incertitude (polygones noirs), contrairement aux points de fuite orthogonalisés (triangles roses). En haut à gauche : la photo P1030004 avec les segments annotés et les lignes d'horizon fournies. Les points de fuite et les régions d'incertitudes associés aux segment verts, rouges et bleus sont représentés respectivement en haut à droite, en bas à gauche et en bas à droite.

tenu des diverses imprécisions des modèles utilisés, nous proposons des polygones modélisant les régions d'incertitude des points de fuite calculés à partir des segments de la scène. Nous pensons que l'utilisation de données inertielles permet de faciliter le calcul des points de fuite avec des résultats plus robustes, notamment lorsqu'un grand nombre de segments aberrants est présent. Le Toulouse Vanishing Points Dataset peut être téléchargé sur cette page web : <http://ubee.enseeiht.fr/tvdpd>

## Références

- [1] D G Aguilera, J Gómez Lahoz, and J Finat Codes. A new method for vanishing points detection in 3D reconstruction from a single view. In *Proceedings of the ISPRS Working Group V/4 Workshop*, 2005.
- [2] M.E. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, volume 2, pages 282–289. IEEE Comput. Soc., 2000.
- [3] Michel Antunes and Joao P Barreto. A Global Approach for the Detection of Vanishing Points and Mutually Orthogonal Vanishing

- Directions. *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2013)*, pages 1336–1343, June 2013.
- [4] Olga Barinova, Victor Lempitsky, Elena Tretyak, and Pushmeet Kohli. Eurasian Cities Dataset. <http://graphics.cs.msu.ru/en/research/projects/msr/geometry>, 2010.
- [5] S Barnard. Interpreting perspective images. *Artificial Intelligence*, 21(4) :435–462, November 1983.
- [6] Jean Charles Bazin, Cedrick Démonceaux, Pascal Vasseur, K. Ikeuchi, and Marc Pollefeys. Globally optimal line clustering and vanishing point estimation in Manhattan world. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2012)*, number 1, pages 638–645. IEEE, June 2012.
- [7] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry : Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd edition, 2008.
- [8] J. Y. Bouguet. Camera calibration toolbox for Matlab, 2008.
- [9] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2) :127–139, March 1990.
- [10] Jiwon Choi, Wonjun Kim, Haejung Kong, and Changick Kim. Real-time vanishing point detection using the Local Dominant Orientation Signature. In *Proceedings of the 2011 3DTV Conference : The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE, 2011.
- [11] R.T. Collins and R.S. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *Proceedings of the 1990 IEEE International Conference on Computer Vision (ICCV 1990)*, pages 400–403. IEEE Comput. Soc. Press, 1990.
- [12] James M. Coughlan and A. L. Yuille. Manhattan world : Orientation and outlier detection by bayesian inference. *Neural Comput.*, 15(5) :1063–1088, May 2003.
- [13] W.D. Curtis, A.L. Janin, and K. Zikan. A note on averaging rotations. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pages 377–385. IEEE, 1993.
- [14] Patrick Denis, James H. Elder, and Francisco J. Estrada. Efficient Edge-Based Methods for Estimating Manhattan Frames in Urban Imagery. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Proceedings of the 2008 European Conference on Computer Vision (ECCV 2008)*, pages 197–210. Springer Berlin Heidelberg, 2008.
- [15] Patrick Denis, James H. Elder, and Francisco J. Estrada. York Urban Line Segment Database. <http://www.elderlab.yorku.ca/YorkUrbanDB/>, 2008.
- [16] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and Modeling of Inertial Sensors Using Allan Variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1) :140–149, January 2008.
- [17] P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1280–1286, Nov 2013.
- [18] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD : a Line Segment Detector. *Image Processing On Line*, 2 :35–55, March 2012.
- [19] Christopher Ham, Simon Lucey, and Surya Singh. Hand Waving Away Scale. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proceedings of the 2014 European Conference on Computer Vision (ECCV 2014)*, volume 8692 of *Lecture Notes in Computer Science*, pages 279–293, Cham, 2014. Springer International Publishing.
- [20] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. *2009 IEEE 12th International Conference on Computer Vision*, pages 1849–1856, September 2009.
- [21] Antonio Torralba, Joseph J. Lim, Hamed Pirsiavash. Parsing ikea objects : Fine pose estimation. In *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV 2013)*, 2013.
- [22] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1) :35, 1960.
- [23] J Košecká and W Zhang. Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding*, (February 2005), 2005.
- [24] Jana Košecká and Wei Zhang. Video Compass. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Proceedings of the 2002 European Conference on Computer Vision (ECCV2002)*, pages 476–490. Springer Berlin Heidelberg, 2002.
- [25] Bo Li, Kun Peng, Xianghua Ying, and Hongbin Zha. PKU Campus Database. <http://www.cis.pku.edu.cn/vision/vpddetection/>, 2012.
- [26] Bo Li, Kun Peng, Xianghua Ying, and Hongbin Zha. Vanishing point detection using cascaded 1D Hough Transform from single images. *Pattern Recognition Letters*, 33(1) :1–8, January 2012.
- [27] Branislav Micusik, Horst Wildenauer, and Jana Kosecka. Detection and matching of rectilinear structures. In *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008)*, pages 1–7. IEEE, June 2008.
- [28] Marcos Nieto and Luis Salgado. Real-time robust estimation of vanishing points through nonlinear optimization. In Nasser Kehtarnavaz and Matthias F. Carlssohn, editors, *Proceedings of SPIE 7724, Real-Time Image and Video Processing 2010*, pages 772402–772402–14, April 2010.
- [29] Srikumar Ramalingam, Jaishanker K Pillai, Arpit Jain, and Yuichi Taguchi. Manhattan Junction Catalogue for Spatial Reasoning of Indoor Scenes. In *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV2013)*, pages 3065–3072. IEEE, June 2013.
- [30] Carsten Rother. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10) :647–655, August 2002.
- [31] G. Schindler and F. Dellaert. Atlanta world : an expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, volume 1, pages 203–209. IEEE, 2004.
- [32] Alexander G. Schwing, Sanja Fidler, Marc Pollefeys, and Raquel Urtasun. Box in the Box : Joint 3D Layout and Object Reasoning from Single Images. In *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV2013)*, pages 353–360. IEEE, December 2013.
- [33] JA Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3) :0–6, 1999.
- [34] Petri Tanskanen, Kalin Kolev, Lorenz Meier, Federico Camposeco, Olivier Saurer, and Marc Pollefeys. Live metric 3d reconstruction on mobile phones. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, Washington, DC, USA, 2013. IEEE Computer Society.
- [35] Jean-Philippe Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *Proceedings of the 2009 IEEE International Conference on Computer Vision (ICCV2009)*, pages 1250–1257. IEEE, September 2009.
- [36] Elena Tretyak, Olga Barinova, Pushmeet Kohli, and Victor Lempitsky. Geometric Image Parsing in Man-Made Environments. *International Journal of Computer Vision*, 97(3) :305–321, September 2011.
- [37] H. Wildenauer and A. Hanbury. Robust camera self-calibration from monocular images of Manhattan worlds. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2012)*, pages 2831–2838. IEEE, June 2012.
- [38] Horst Wildenauer and Markus Vincze. Vanishing Point Detection in Complex Man-made Worlds. In *Proceedings of the 2007 International Conference on Image Analysis and Processing (ICIAP 2007)*, pages 615–622. IEEE, September 2007.

- [39] Yiliang Xu, Sangmin Oh, and Anthony Hoogs. A Minimum Error Vanishing Point Detection Approach for Uncalibrated Monocular Images of Man-Made Environments. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2013)*, pages 1376–1383. IEEE, June 2013.