



HAL
open science

Exploration réaliste et pertinente d'un nuage de points 3D dense et coloré

Zaynab Habibi, Guillaume Caron, El Mustapha Mouaddib

► **To cite this version:**

Zaynab Habibi, Guillaume Caron, El Mustapha Mouaddib. Exploration réaliste et pertinente d'un nuage de points 3D dense et coloré. Journées francophones des jeunes chercheurs en vision par ordinateur, ORASIS'15, Jun 2015, Amiens, France. hal-01161823

HAL Id: hal-01161823

<https://hal.science/hal-01161823>

Submitted on 9 Jun 2015

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

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

Exploration réaliste et pertinente d'un nuage de points 3D dense et coloré

Zaynab Habibi

Guillaume Caron

El Mustapha Mouaddib

Université de Picardie Jules Verne, Laboratoire MIS

33 rue Saint Leu, 80039 Amiens Cedex 1, France
zaynab.habibi, guillaume.caron, mouaddib@u-picardie.fr

Résumé

Dans un environnement 3D composé d'un nuage de points dense, la navigation sans assistance est une tâche complexe décourageant les utilisateurs. Le but de ce travail est de proposer un outil d'aide à la navigation virtuelle. Plus précisément, cet outil permettra de guider automatiquement la caméra de façon à générer des mouvements pertinents et réalistes. Nous avons modélisé ce problème sous la forme d'une tâche principale et de contraintes. La tâche principale décrit la pertinence, formulée sous l'hypothèse que l'utilisateur est toujours attiré par les endroits riches en information visuelle. Cette formulation s'appuie sur l'entropie de Shannon. Par la suite, pour garantir un mouvement réaliste, deux contraintes ont été ajoutées : l'évitement d'obstacles et la régularisation du mouvement dans l'image. Dans ce travail, nous proposons un mécanisme d'optimisation hiérarchique en deux étapes qui permet de combiner la tâche et les contraintes en un seul problème et de le résoudre. Notre approche a été testée en navigation intérieure et extérieure, d'abord sur un modèle 3D simple afin de valider les critères retenus, puis sur deux modèles 3D très complexes : un environnement urbain et la cathédrale d'Amiens.

Mots Clef

Navigation automatique, entropie, optimisation.

Abstract

In a 3D dense points cloud model, virtual tour without assistance is a complex and difficult task discouraging users from doing so. The aim of this work is to achieve a virtual navigation support tool. It will help users to perform virtual tours to explore the 3D models. In particular, the tool will allow to guide the camera automatically. We assume that the user is attracted by rich information areas in the model. This important assumption will be modelled by entropy. Secondly, in order to achieve a realistic automatic navigation we must avoid obstacles, ensure a relevant camera orientation during its motion and regulate the visual movement in the produced image. In this paper, we propose a solution to this problem based on a hierarchical algorithm, which combines the main task

to be achieved and the realistic constraints. We validate the system on different complex 3D models : lab, urban environment and a cathedral.

Keywords

Automatic navigation, entropy, optimization.

1 Introduction

Ces dernières années, les applications virtuelles 3D comme la visite des monuments architecturaux et les jeux vidéo ont reçu une attention particulière dans le domaine de l'animation par les chercheurs. Cependant, les logiciels existants pour l'exploration d'un environnement virtuel 3D montrent quelques inconvénients, particulièrement pour les utilisateurs non initiés comme la difficulté de trouver un point de vue pertinent et surtout les mouvements irréalistes de la caméra.

Dans ce travail¹, on souhaite réaliser une navigation automatique pertinente, du point de vue du contenu perçu dans l'image à l'instar du résultat illustré en Figure 1. La caméra virtuelle se déplace dans la scène 3D dans le but de visualiser les informations les plus pertinentes.

En contrôle automatique d'une caméra virtuelle on retrouve deux familles de méthodes : les méthodes de planification de trajectoires et les méthodes basées images. Pour les méthodes de planification de trajectoires, la première étape est de définir des points de passage manuellement ou automatiquement en échantillonnant l'espace de configuration de la caméra. Généralement, des primitives basées sur la théorie de l'information (spécialement l'entropie) sont utilisées pour la sélection de points de vue pertinents pour imiter le comportement d'un utilisateur attiré par les parties les plus intéressantes d'une scène 3D.

1. Une version anglophone de cet article a été publiée dans les actes de l'atelier "E-heritage" de la conférence ACCV(Asian Conference on Computer Vision 2014 à Singapour) [1].



FIGURE 1 – Chemin visuel maximisant l'information pertinente dans l'image.

Dans un environnement statique, la qualité d'un point de vue peut être évaluée en calculant l'entropie géométrique visuelle. Dans [2], les auteurs ont considéré un modèle surfacique pour calculer la distribution de probabilité de l'entropie et ils ont utilisé la surface relative d'une face projetée sur une sphère centrée sur le point de vue (rapport entre la surface de la face et la surface totale du modèle). L'entropie est maximale quand on peut visualiser toutes les surfaces du modèle 3D à partir d'un certain point de vue. Dans [3], l'entropie a été exploitée pour trouver le point de vue qui permet de maximiser la luminosité. Un algorithme adaptatif de recherche utilise l'intensité des pixels pour calculer la distribution de probabilité de l'entropie. Dans un modèle 3D constitué de plusieurs cellules [4], l'entropie visuelle géométrique introduite par les auteurs de [2] a été exploitée pour calculer des points de vue pertinents. Ensuite, un algorithme de backtracking est utilisé pour explorer toutes les cellules du modèle. La seconde étape des algorithmes de planification de trajectoire est de connecter les points de vue qui ont été déterminés soit manuellement soit en se basant sur une entropie géométrique [2] voire photométrique [3]. On retrouve parmi les techniques de planification, l'algorithme A^* [5], le problème du voyageur de commerce [6], les feuilles de route [7][8], les méthodes de décomposition spatiale [9][10] et les méthodes basées sur les champs de potentiel [11] [12] qui peuvent être utilisées pour connecter les points de vue. Finalement, la trajectoire de la caméra virtuelle est construite en interpolant cet ensemble de points de vue en utilisant des courbes splines par exemple [13] afin de déterminer les points de vue intermédiaires.

Dans les méthodes basées image, la trajectoire de la caméra est construite en optimisant une fonction de coût qui est définie à partir de primitives de l'image. La mise à jour des six degrés de liberté (d.d.l) de la caméra est réalisée à l'aide d'un processus itératif. Dans ce contexte, Courty et al. [14] ont proposé une méthode basée sur l'asservissement visuel. Cette méthode effectue une tâche de positionnement dans l'environnement en considérant des contraintes sur la caméra. Elle permet d'une part de suivre un objet et d'une autre part de fournir une assistance pour créer une trajectoire de caméra en utilisant des primitives cinématographiques. Les méthodes de planification de trajectoire précédemment présentées opèrent d'une manière séquentielle. Elles commencent par exploiter l'entropie visuelle pour déterminer les meilleurs points de vue dans un processus préliminaire, par la suite le chemin est calculé en reliant les points de vue sélectionnés. Ces approches peuvent être accompagnées d'une contrainte d'évitement d'obstacles [14] ou non. Dans ce cas, la pertinence n'est relative qu'aux points de vue et non aux mouvements de caméra qui permet de les relier. D'un autre côté, la navigation automatique basée asservissement visuel présentée dans [14] a comme tâche principale le suivi d'une cible (garder la cible centrée dans l'image) dont la trajectoire a été générée à l'aide d'une méthode de planification de tra-

jectoire. Dans cette méthode, une étape de pré-traitement en hors-ligne est obligatoire pour générer la trajectoire de la caméra. Pour résoudre ce problème, nous utilisons les principes de l'asservissement visuel, pour générer la trajectoire de la caméra dans un processus en-ligne en réalisant un contrôle basé entropie combiné avec d'autres contraintes décrites dans ce qui suit.

1.1 Stratégie générale

On propose de calculer l'entropie visuelle photométrique à partir de l'intensité des pixels de l'image. Ceci la rend dépendante du point de vue et non du type du modèle 3D (nuage de points, maillage). Le contrôle automatique de la caméra est mis en œuvre par un asservissement visuel virtuel, en utilisant comme primitive l'entropie photométrique. En concevant la loi de commande qui maximise l'entropie, cette tâche permet de calculer la direction du mouvement sans aucune étape de planification ou de connexion entre des points de vue choisis au préalable. Dans le but d'assurer une orientation pertinente (éviter les rotations autour de l'axe optique de la caméra), on utilise cinq degrés de liberté opérationnels pour la caméra (les trois translations, le "pan" et le "tilt") au lieu des six d.d.l cartésiens généralement considérés en asservissement visuel. La caméra virtuelle doit éviter les obstacles durant son mouvement. Beaucoup de travaux en contrôle de caméra se sont intéressés à cette problématique comme les méthodes de lancer de rayons [15][16][17], les méthodes de volumes englobants [14] [18] et les méthodes basées image [19]. En ce qui nous concerne, nous avons adapté une méthode proposée par Courty et al. [14] qui consiste à maximiser une fonction de coût qui tend vers l'infini quand la distance entre la caméra et l'obstacle est nulle. Dans notre approche, on maximise la distance moyenne entre la caméra et tous les sommets du modèle qui se trouvent dans un rayon fixé et centré sur la caméra virtuelle. La maximisation de l'entropie photométrique et de la distance entre la caméra et les obstacles sont combinées en utilisant une loi de contrôle hybride qui consiste à empiler les deux tâches dans un même système en les pondérant [20]. Cela permet d'éviter les changements de direction soudains à chaque fois qu'un obstacle est détecté. Quand la loi de contrôle basée sur l'entropie photométrique et l'évitement d'obstacles converge, la pertinence devient quasiment statique (la même quantité d'information dans l'image). Cependant, pour que le mouvement soit dynamiquement pertinent, il faut éviter les sauts brusques entre deux images successives (discontinuité du mouvement dans l'image). Pour résoudre ce problème de discontinuité du mouvement dans l'image, nous avons proposé une loi de contrôle, basée sur le flot optique, destinée à régulariser le mouvement dans l'image. Pour prendre en compte la maximisation de l'entropie, l'évitement des obstacles et la régularisation du flot optique, nous avons proposé une loi de contrôle hiérarchique en deux étapes, qui combine ces critères et qui estime la direction et l'amplitude du mouvement. En plus, nous avons

implémenté une alternance entre le processus de maximisation et de minimisation de l'entropie pour éviter de stagner dans un maximum local. On commence par le processus de maximisation de l'entropie photométrique pour amener la caméra vers les parties les plus pertinentes du modèle, ensuite un processus de minimisation de l'entropie est déclenché pour changer de zone. Cette alternance permet de générer une plus grande trajectoire. La durée de chaque étape peut être définie par l'utilisateur.

1.2 Contributions

Dans le cadre de l'approche proposée, les contributions sont :

- L'utilisation de l'entropie pour assurer une navigation pertinente.
- La génération de mouvements réalistes en ajoutant des contraintes.
- La mise au point d'une méthode hiérarchique d'optimisation pour combiner l'optimisation de l'entropie, l'évitement d'obstacles et la régularisation du mouvement dans l'image.
- La mise en œuvre d'une navigation intérieure et extérieure sur différents modèles 3D complexes.

La suite du papier présente une description de l'approche proposée et des résultats expérimentaux avant de conclure.

2 Tâche principale et contraintes

2.1 L'entropie photométrique

On considère $I(\mathbf{r})$ l'image à la pose $\mathbf{r} = (t_X, t_Y, t_Z, \theta u_X, \theta u_Y, \theta u_Z)^\top$, où les rotations sont représentées par un angle θ et un vecteur unitaire (axe de rotation) $\mathbf{u} = [u_X, u_Y, u_Z]^\top$. L'entropie photométrique d'une image est donnée par :

$$E(\mathbf{r}) = - \sum_i P_i(\mathbf{r}) \log_2(P_i(\mathbf{r})) \quad (1)$$

tel que i est un niveau de gris d'un pixel dans l'image, $i \in \llbracket 0; 255 \rrbracket$ et $P_i(\mathbf{r})$ la probabilité que l'intensité i existe dans l'image I . Pour un souci de lisibilité $P_i(\mathbf{r})$ est remplacée par P_i dans ce qui suit.

La distribution de probabilité est obtenue en utilisant un histogramme d'intensités normalisé. En s'inspirant de [21] dans le cas du calcul de l'information mutuelle entre deux images dans un autre contexte, nous approximations l'histogramme normalisé à l'aide d'une B-Spline d'ordre 2 [22] notée ϕ . Ceci permet une optimisation du second ordre en rendant l'entropie photométrique deux fois différentiable par rapport à la pose \mathbf{r} . La probabilité est exprimée par :

$$P_i = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \phi(i - I(\mathbf{u}, \mathbf{r})) \quad (2)$$

$\mathbf{u} = (u, v)$ sont les coordonnées d'un pixel \mathbf{u} , $I(\mathbf{u}, \mathbf{r})$ est l'intensité à la pose courante \mathbf{r} et $N_{\mathbf{u}}$ le nombre de pixels dans l'image.

2.2 Contrainte d'évitement d'obstacles

Nous considérons tous les sommets qui se trouvent à une distance d'un mètre de la caméra virtuelle. La fonction de coût définie dans [14] tend vers l'infini quand la distance entre la caméra et l'obstacle est nulle. Chaque sommet est considéré comme un obstacle, le but est donc de maximiser la distance entre la caméra et les sommets en minimisant :

$$O(\mathbf{r}) = \sum_k \frac{1}{2\|\mathbf{p}_c - \mathbf{p}_{o_k}\|^2} \quad (3)$$

tel que k est l'indice d'un sommet. $\mathbf{p}_c = (t_X, t_Y, t_Z)^\top$ est la position de la caméra et $\mathbf{p}_{o_k} = (X_{o_k}, Y_{o_k}, Z_{o_k})^\top$ est la position du sommet dans le repère monde.

2.3 Description du flot optique

Cette contrainte assure un mouvement fluide dans l'espace image. L'approche consiste à fixer une amplitude du flot optique entre les images $\Delta \mathbf{u}_{des} = [\Delta u_{des}, \Delta v_{des}]^\top$, ensuite l'équation de conservation des intensités est utilisée pour lier $\Delta \mathbf{u}_{des}$ et la pose de la caméra \mathbf{r} :

$$I(\mathbf{u}, \mathbf{r}) = I(\mathbf{u} + \Delta \mathbf{u}_{des}, \mathbf{r} + \Delta \mathbf{r}) \quad (4)$$

3 Processus de résolution

La tâche principale et les contraintes sont non linéaires par rapport à la pose de la caméra \mathbf{r} . Un processus de résolution non linéaire itératif est appliqué. La linéarisation est effectuée à l'aide du développement de Taylor. Nous considérons d'abord le cas général d'une loi de contrôle pour les six d.d.l cartésiens de la caméra, et nous présenterons la restriction à cinq degrés opérationnels à la fin de cette section. Le processus itératif consiste à estimer l'incrément de pose $\Delta \mathbf{r}$ afin de modifier la pose courante \mathbf{r}_k (itération k) :

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \Delta \mathbf{r}_k \quad (5)$$

3.1 Critère de l'entropie photométrique

La maximisation de l'entropie photométrique est équivalente à la minimisation de son opposé $-E(\mathbf{r}_k)$. Ainsi, la recherche du maximum revient à calculer les racines de la dérivée de $-E(\mathbf{r}_k)$, en appliquant le développement de Taylor au 1^{er} en \mathbf{r}_k , on obtient :

$$\frac{\partial(-E(\mathbf{r}_k))}{\partial \mathbf{r}} + \left(\frac{\partial^2(-E(\mathbf{r}_k))}{\partial \mathbf{r}^2} \right) \Delta \mathbf{r}_k = 0 \quad (6)$$

Pour résoudre cette équation, le gradient et le hessien de la fonction de coût doivent être calculés. En utilisant l'équation (1) le gradient de l'entropie photométrique est donné par :

$$\frac{\partial}{\partial \mathbf{r}}(-E(\mathbf{r}_k)) = \sum_i \frac{\partial P_i}{\partial \mathbf{r}} (1 + \log(P_i)) \quad (7)$$

Avec l'expression de la probabilité exprimée dans l'équation (2), on obtient :

$$\frac{\partial P_i}{\partial \mathbf{r}} = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \left(-\frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_k))}{\partial (i - I(\mathbf{u}, \mathbf{r}_k))} \nabla I \mathbf{L}_{\mathbf{u}} \right) \quad (8)$$

où $\nabla I = (\nabla_u I, \nabla_v I)$ est le gradient de l'image $I(\mathbf{r})$ et \mathbf{L}_u la matrice d'interaction. Elle lie les variations dans l'image aux variations de la pose de la caméra au pixel $\mathbf{u} = (u, v)$, et se définit par :

$$\mathbf{L}_u = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \mathbf{L}_x \quad (9)$$

avec \mathbf{L}_x la matrice définie dans [18] et $\mathbf{x} = (x, y)$ issu des relations :

$$\begin{cases} u = \alpha_u x + u_0 \\ v = \alpha_v y + v_0 \end{cases} \quad (10)$$

Dans l'équation (10) $\alpha_u, \alpha_v, u_0, v_0$ les paramètres intrinsèques de la caméra.

Le hessien de l'entropie photométrique est donné par :

$$\frac{\partial^2 (-E(\mathbf{r}_k))}{\partial \mathbf{r}^2} = \sum_i \frac{\partial P_i}{\partial \mathbf{r}} \frac{\partial P_i^T}{\partial \mathbf{r}} \frac{1}{P_i} + \frac{\partial^2 P_i}{\partial \mathbf{r}^2} (1 + \log(P_i)) \quad (11)$$

En utilisant la première dérivée de probabilité $\frac{\partial P_i}{\partial \mathbf{r}}$, on calcule sa dérivée seconde $\frac{\partial^2 P_i}{\partial \mathbf{r}^2}$ ce qui nous donne :

$$\frac{1}{N_u} \sum_u \left(\frac{\partial^2 \phi(i - I(\mathbf{u}, \mathbf{r}_k))}{\partial (i - I(\mathbf{u}, \mathbf{r}_k))^2} \mathbf{L}_I^T \mathbf{L}_I - \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_k))}{\partial (i - I(\mathbf{u}, \mathbf{r}_k))} \mathbf{H}_I \right) \quad (12)$$

Dans cette dernière équation, $\mathbf{L}_I = \nabla I \mathbf{L}_u$ et,

$$\mathbf{H}_I = \mathbf{L}_u^T \nabla^2 I \mathbf{L}_u + \nabla_u I \mathbf{H}_u + \nabla_v I \mathbf{H}_v \quad (13)$$

tel que $\nabla^2 I = \begin{pmatrix} \nabla I_{uu} & \nabla I_{uv} \\ \nabla I_{vu} & \nabla I_{vv} \end{pmatrix}$ est le gradient de l'image gradient. \mathbf{H}_u et \mathbf{H}_v sont les hessiens de taille 6×6 des deux coordonnées du point \mathbf{u} [23].

3.2 Contrainte d'évitement d'obstacles

Le but ici est de minimiser l'équation (3) ce qui est équivalent à calculer les racines de $O(\mathbf{r})$. En utilisant le développement de Taylor au 1^{er} ordre en \mathbf{r}_k , nous avons :

$$O(\mathbf{r}_k) + \left(\frac{\partial(O(\mathbf{r}_k))}{\partial \mathbf{r}} \right) \Delta \mathbf{r}_k \quad (14)$$

Nous voulons que la caméra virtuelle évite les obstacles (les sommets) durant sa navigation dans le modèle 3D. De ce fait, seule la position de la caméra est considérée et non son orientation. On modélise la caméra par son centre optique, dans son repère $\mathbf{p}_c = (0, 0, 0)^T$, donc :

$$\frac{\partial(O(\mathbf{r}_k))}{\partial \mathbf{r}} = \sum_k 4O(\mathbf{r}_k)^2 (X_{o_k}, Y_{o_k}, Z_{o_k}, 0, 0, 0) \quad (15)$$

3.3 Contrainte basée sur le flot optique

A partir de l'équation (4) et en utilisant encore une fois le développement de Taylor au 1^{er} ordre, nous avons :

$$I(\mathbf{u} + \Delta \mathbf{u}_{des}, \mathbf{r}_k + \Delta \mathbf{r}_{k_f}) = I(\mathbf{u}, \mathbf{r}_k) + \frac{\partial I}{\partial \mathbf{u}} \Delta \mathbf{u}_{des} + \frac{\partial I}{\partial \mathbf{r}} \Delta \mathbf{r}_{k_f} \quad (16)$$

tel $\Delta \mathbf{r}_{k_f}$ est l'incrément de pose final. La substitution de (4) dans (16) donne :

$$\frac{\partial I}{\partial \mathbf{u}} \Delta \mathbf{u}_{des} + \frac{\partial I}{\partial \mathbf{u}} \mathbf{L}_u \Delta \mathbf{r}_{k_f} = 0 \quad (17)$$

3.4 Algorithme hiérarchique

Pour combiner la tâche principale (maximisation de l'entropie photométrique) et les autres contraintes, on pourrait faire une résolution globale en empilant toutes les équations dans une seule matrice. Cependant, le résultat obtenu n'est pas satisfaisant parce que les contraintes ne sont pas toujours toutes respectées. Dans les méthodes basées image, on trouve le formalisme de redondance qui est souvent utilisé pour faire du multi-tâches. Cette technique est réalisée en affectant à chaque tâche un certain nombre de d.d.l. Dans [14], les auteurs ont exploité cette technique pour combiner la tâche principale de suivi d'objet qui ne requiert pas tous les d.d.l de la caméra et une seconde tâche (gestion des occultations ou de l'éclairage, ...). Dans le formalisme de redondance, il n'est pas possible qu'un d.d.l soit exploité par deux tâches différentes. Il ne peut donc pas être exploité dans notre cas car la tâche principale requiert tous les d.d.l de la caméra. C'est pourquoi, nous avons proposé un nouvel algorithme hiérarchique de contrôle de caméra. L'approche est réalisée en deux étapes comme illustré sur la Figure 2a. Elle prend en entrée un modèle 3D et une pose initiale de la caméra \mathbf{r} . On commence par calculer la direction du mouvement à l'aide de la loi de contrôle hybride entropie-obstacles qui maximise l'entropie et la distance entre la caméra et les obstacles voisins. Ensuite, on ajuste l'amplitude du mouvement qui doit générer un flot optique globalement constant afin d'avoir un mouvement régulier dans l'image. La trajectoire de la caméra est générée à l'aide d'un processus d'optimisation itératif qui prend en compte ces deux étapes.

Étape 1 : estimation de la direction du mouvement

Nous l'obtenons en empilant directement les équations (6) et (14). L'estimation de $\Delta \mathbf{r}$ est donnée par :

$$\Delta \mathbf{r}_k = \begin{pmatrix} \frac{\partial^2 (-E(\mathbf{r}_k))}{\partial \mathbf{r}^2} \\ \frac{\partial(O(\mathbf{r}_k))}{\partial \mathbf{r}} \end{pmatrix}^+ \begin{pmatrix} \frac{\partial}{\partial \mathbf{r}} (-E(\mathbf{r}_k)) \\ O(\mathbf{r}_k) \end{pmatrix} \quad (18)$$

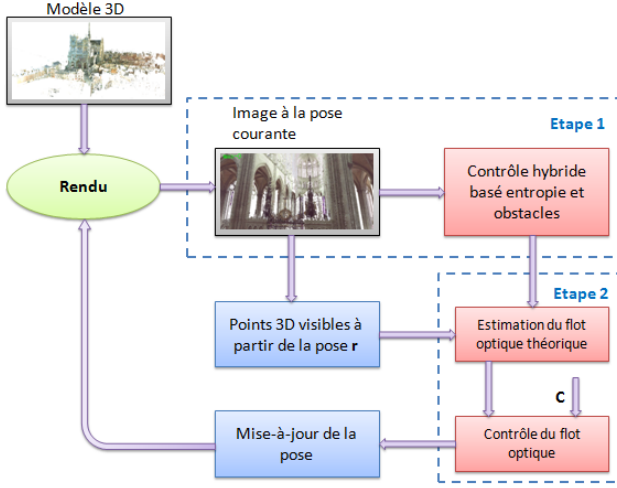
Dans ce qui suit on pose $\mathbf{H}^+ = \begin{pmatrix} \frac{\partial^2 (-E(\mathbf{r}_k))}{\partial \mathbf{r}^2} \\ \frac{\partial(O(\mathbf{r}_k))}{\partial \mathbf{r}} \end{pmatrix}^+$ et

$\mathbf{g} = \begin{pmatrix} \frac{\partial}{\partial \mathbf{r}} (-E(\mathbf{r}_k)) \\ O(\mathbf{r}_k) \end{pmatrix}$. A partir de l'incrément de pose

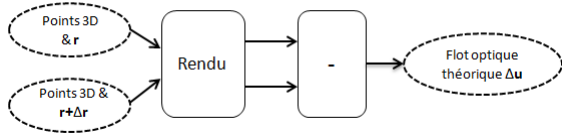
$\Delta \mathbf{r}_k$, nous utilisons l'application exponentielle du groupe spécial euclidien SE(3) pour mettre à jour la pose courante \mathbf{r}_k (équation (5) et ainsi obtenir la matrice de transformation associée.

Étape 2 : Estimation finale de la pose Cette étape régularise dans l'image l'amplitude du mouvement, en procédant en deux temps :

1. On calcule le flot optique théorique Δu et Δv comme illustré sur la Figure 2b, où $I(\mathbf{r})$ et $I(\mathbf{r} + \Delta \mathbf{r})$ sont les images de synthèse rendues par le moteur graphique à partir du modèle 3D.



(a) Processus de contrôle de caméra.



(b) Calcul du flot optique théorique (calcul de la différence entre les coordonnées des points des images rendues à la pose \mathbf{r} et $\mathbf{r} + \Delta\mathbf{r}$).

FIGURE 2 – Vue générale de l’algorithme.

- On fixe une constante C comme amplitude désirée du flot optique global.

Le flot désiré est calculé comme suit, en chaque pixel :

$$\Delta u_{des} = \frac{C \cdot \Delta u}{\sqrt{\Delta u^2 + \Delta v^2}} \quad (19)$$

$$\Delta v_{des} = \frac{C \cdot \Delta v}{\sqrt{\Delta u^2 + \Delta v^2}} \quad (20)$$

tel que C est une constante qui correspond à l’amplitude moyenne du flot optique. Le déplacement final de la caméra est calculé à partir de l’équation (17) :

$$\Delta \mathbf{r}_{k_f} = -\mathbf{L}_{\mathbf{u}}^+ \Delta \mathbf{u}_{des} \quad (21)$$

Où $\mathbf{L}_{\mathbf{u}}^+$ est la pseudo inverse de $\mathbf{L}_{\mathbf{u}}$.

3.5 Gestion des rotations

La navigation automatique de la caméra dans le modèle 3D génère des mouvements de roulis irréalistes autour de l’axe optique de la caméra. C’est pourquoi, nous procédons à un traitement supplémentaire pour contraindre le mouvement de la caméra à cinq degrés opérationnels : les trois translations $(t_X, t_Y, t_Z)^T$ et les deux rotation $(\theta_p, \theta_t)^T$ correspondant au panorama et à l’inclinaison au lieu des 6 d.d.l cartésiens.

Le comportement de la caméra ressemble à celui d’un robot virtuel avec cinq d.d.l, où les trois translations se font

dans le repère monde \mathcal{R}_w et les deux rotations dans le repère caméra \mathcal{R}_c successivement. On note par \mathbf{q}_k la position du robot à l’itération k .

Le Jacobien du robot ${}^c \mathbf{J}_c$ dans le repère \mathcal{R}_c est donné par :

$${}^c \mathbf{J}_c = {}^c \mathbf{V}_w {}^w \mathbf{J}_w \quad (22)$$

avec ${}^w \mathbf{J}_w$ le jacobien du robot exprimé dans le repère \mathcal{R}_w et ${}^c \mathbf{V}_w$ est une matrice de transformation cinématique de \mathcal{R}_w à \mathcal{R}_c .

Ainsi, l’incrément de pose calculé par l’équation (18) devient :

$$\dot{\mathbf{q}}_k = (\mathbf{H}^c \mathbf{J}_c)^+ \mathbf{g} \quad (23)$$

$\dot{\mathbf{q}}_k$ représente le vecteur à cinq dimensions de l’incrément de pose. Le même raisonnement est appliquée pour la régularisation du flot (équation (21)) :

$$\dot{\mathbf{q}}_{k_f} = (\mathbf{L}_{\mathbf{u}}^c \mathbf{J}_c)^+ \Delta \mathbf{u}_{des} \quad (24)$$

$\dot{\mathbf{q}}_{k_f}$ représente le vecteur à cinq dimensions de l’incrément de pose final.

La matrice de transformation est calculée à partir du vecteur de translation ${}^c \mathbf{t}_w$ et de la matrice de rotation ${}^c \mathbf{R}_w$, tel que ${}^c \mathbf{R}_w = R_X(\theta_t) R_Z(\theta_p)$. Le vecteur de translation est donné par ${}^c \mathbf{t}_w = -{}^c \mathbf{R}_w^c \mathbf{t}_w$ avec ${}^c \mathbf{t}_w = (\mathbf{q}_{k+1}[0], \mathbf{q}_{k+1}[1], \mathbf{q}_{k+1}[2])^T$.

4 Résultats expérimentaux

Nous avons testé l’approche proposée sur trois modèles 3D composés de nuages de points colorés. Notre algorithme est compatible avec des modèles de large échelle et complexes. L’approche est validée sur un modèle 3D et un modèle partiel de la cathédrale d’Amiens. Ces deux environnements sont obtenus par relevés laser (Faro Focus 3D). Comme nous pouvons le voir sur les images rendues, l’environnement n’a pas été scanné dans sa totalité (zones blanches dans les images). Chaque environnement utilisé, compte environ cent millions de points 3D. Pour la visualisation des modèles, nous avons utilisé le moteur graphique Ogre 3D [26].

4.1 Maximisation de l’entropie photométrique

Environnement basique. Dans cette expérience nous montrons le comportement de la maximisation de l’entropie photométrique. Nous allons commencer par appliquer cette loi de contrôle sur un modèle simple : un mur blanc et un poster accroché. En appliquant la maximisation de l’entropie photométrique, nous remarquons que le poster devient de plus en plus présent dans l’image au fur et à mesure du mouvement de la caméra (Figure 3). Ceci représente un premier résultat expérimental qui permet de valider l’utilisation de l’entropie photométrique comme primitive visuelle permettant de maximiser l’information dans l’image.

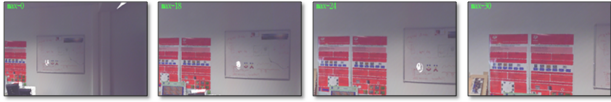


FIGURE 3 – Le trajectoire visuelle de la caméra en maximisant l’information pertinente dans l’image.

Environnement complexe. Dans un autre test, nous appliquons la maximisation de l’entropie en navigation extérieure sur le modèle de la cathédrale. Nous remarquons qu’à partir d’une position initiale où le modèle 3D ne remplit pas totalement l’image (Figure 4), la loi de contrôle basée entropie contraint la caméra à inclure le plus d’informations dans son champs de vue. La Figure 5 montre l’évolution de la fonction de coût tout au long de la trajectoire.

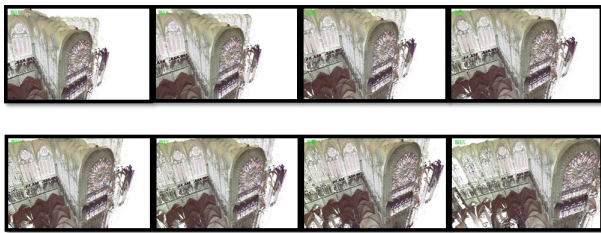


FIGURE 4 – Navigation automatique en maximisant l’entropie photométrique : vue d’extérieur du modèle.

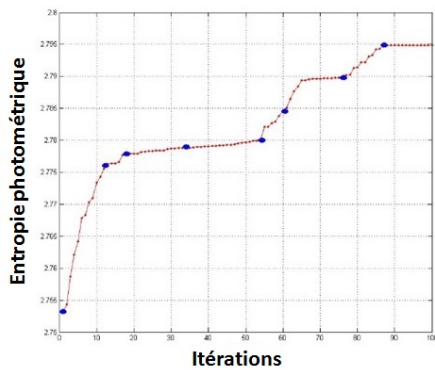


FIGURE 5 – Évolution de l’entropie photométrique. les points bleus correspondent aux images de la Figure 4 (de gauche à droite et de haut en bas).

4.2 Évitement d’obstacles

La contrainte d’évitement d’obstacles est obligatoire en navigation virtuelle car elle évite des comportements irréalistes de la caméra virtuelle comme traverser des murs ou des piliers à l’intérieur du modèle de la cathédrale. La Figure 6 montre le comportement de la caméra suite à la loi de contrôle hybride entropie-obstacles (équation (18)). La Figure 6a montre un extrait des images de synthèse à chaque point de vue la caméra, tandis que la Figure 6b représente les images correspondantes contenant les obstacles (les sommets) détectés à cette pose. Nous avons implémenté une caméra équirectangulaire en utilisant un shader. On remarque, qu’à partir de la quatrième image la

caméra s’éloigne des obstacles, ce qui permet de décroître le nombre de sommets dans l’image équirectangulaire. Dans la Figure 7a, on observe en vert la trajectoire de



(a) Séquence d’images en navigation intérieure.



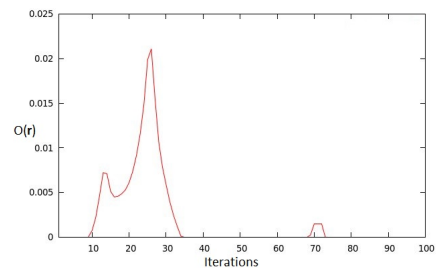
(b) Obstacles (sommets) détectés à partir de chaque pose qui a permis de générer les images de synthèses d’au dessus.

FIGURE 6 – Comportement de la contrainte d’évitement d’obstacles.

la caméra projetée sur le plan 2D du modèle 3D de la cathédrale. Cette trajectoire montre que la caméra passe entre deux piliers pour éviter les collisions. La Figure 7b montre l’évolution de la fonction de coût tout au long du mouvement de la caméra. On remarque que la fonction de coût atteint son maximum quand le nombre d’obstacles détectés dans la sphère (les sommets sont obtenus avec la projection inverse des points de l’image équirectangulaire) augmente (cela correspond à la troisième image de la Figure 6b).



(a) Trajectoire de la caméra sur le plan 2D du modèle 3D de la cathédrale.



(b) Evolution de la fonction de coût d’évitement d’obstacles.

FIGURE 7 – Trajectoire de la caméra.

4.3 Contrôle du flot optique

Dans cette expérience, la caméra est contrôlée en utilisant l’algorithme hiérarchique. Dans le but de mettre en évidence l’importance de cette contrainte, nous avons, à partir d’une même pose initiale, appliqué la loi de contrôle

avec et sans la contrainte basée sur le flot optique. Il est clair, à partir de la Figure 8 qui montre l'évolution de la moyenne de la norme du flot optique calculée pour les deux séquences, que cette contrainte est primordiale dans la mesure où elle permet de lisser le mouvement et d'éviter des sauts brusques. La courbe avec contrôle du flot montre qu'un flot moyen régulier est maintenu en comparaison avec la courbe sans contrôle du flot.

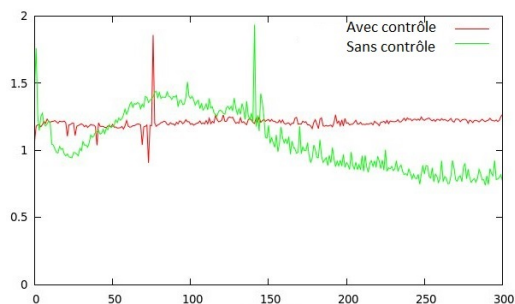


FIGURE 8 – Moyenne de la norme du flot optique pour une séquence d'images considérant la contrainte basée sur le flot (rouge) et une autre sans le considérant (vert).

4.4 Navigation réaliste

Pour obtenir une navigation de longue durée, nous avons alterné la maximisation et la minimisation de l'entropie. Le flot optique est borné afin de passer plus de temps à contempler les informations pertinentes de la scène durant l'étape de maximisation. A l'inverse, durant l'étape de minimisation les mouvements lents ne sont pas pertinents car l'information visuelle devient de plus en plus pauvre. Pour cela, le flot optique est régulé durant cette étape pour avoir un mouvement d'une vitesse constante.

La Figure 9 montre quelques images extraites de la vidéo générée en appliquant l'algorithme proposé. En commençant à partir d'une pose qui contient une partie de la façade, l'algorithme guide la caméra afin d'inclure tout le bâtiment dans son champ de vue. Un autre résultat en navigation intérieure est présenté dans la Figure 10. Une vidéo accompagnant ce papier illustre le comportement de la caméra virtuelle <http://mis.u-picardie.fr/~g-caron/videos/ORASIS2015.mp4>.

5 Conclusions

Nous avons proposé une méthode de contrôle automatique de caméra dans un environnement virtuel constitué d'un nuage de points 3D colorés. L'approche proposée permet de générer des mouvements fluides, réalistes et pertinents. Le contrôle a été réalisé à l'aide d'un algorithme hiérarchique en deux étapes. La direction du mouvement est d'abord déterminée en combinant la tâche de maximisation de l'entropie photométrique et la contrainte d'évitement d'obstacles. Nous avons pu montrer l'intérêt d'utiliser l'entropie pour se diriger vers les endroits intéressants d'une scène donnée. Ensuite, nous avons contrôlé le flot optique dans l'image afin d'éviter les mou-

vements brusques de la caméra. Nous avons montré qu'à partir d'une position initiale pauvre en information, l'algorithme guide la caméra vers des endroits plus intéressants. Cette méthode est générale dans la mesure où on peut intégrer d'autres critères à optimiser.

Il serait intéressant par la suite d'intégrer l'interactivité en ajoutant une autre contrainte à ce mécanisme ce qui permettra d'un côté de remplacer le mécanisme d'alternance entre la maximisation et la minimisation, et d'un autre côté, de rendre possible l'assistance de l'utilisateur dans l'exploration du modèle 3D. Enfin, nous envisageons de réaliser une étude basée sur l'avis de plusieurs utilisateurs afin d'évaluer le système.

Références

- [1] Z. Habibi, G. Caron, E. Mouaddib, "3D model automatic exploration : Smooth and Intelligent Virtual Camera Control", Workshop E-Heritage - Asian Conference on Computer Vision ACCV, Springer, 2014.
- [2] P. Vázquez, M. Feixas, M. Sbert, W. Heidrich, *Viewpoint Selection using Viewpoint Entropy*, Workshop on vision, Modelling and Visualization, vol. 1, pp. 273-280, 2001.
- [3] S. Gumhold, *Maximum entropy light source placement*, Visualization, VIS, IEEE, pp. 275-282, 2002.
- [4] C. Andújar, Carlos and P. Vázquez, and M. Fairén, *Way-Finder : guided tours through complex walkthrough models*, Computer Graphics Forum, Wiley Online Library, Vol. 23, pp. 499-508, 2004.
- [5] I. Yeh, C. Lin, H. Chien, T. Lee and others, *Efficient camera path planning algorithm for human motion overview*, Computer Animation and Virtual Worlds Wiley Online Library. vol. 22, pp. 239-250, 2011.
- [6] E. Serin, S. Hasan Adali and S. Balcisoy, *Automatic path generation for terrain navigation*, Computers & Graphics, Elsevier, vol. 36, pp. 1013-1024, 2012.
- [7] T. Li and C. Cheng, *Real-time camera planning for navigation in virtual environments*, Smart Graphics, Springer, pp. 118-129, 2008.
- [8] B. Salomon, M. Garber, M. Lin and D. Manocha, *Interactive navigation in complex environments using path planning*, Interactive 3D graphics, ACM, pp. 41-50, 2003.
- [9] S. Bandi and D. Thalmann, *Space discretization for efficient human navigation*, Computer Graphics Forum, Wiley Online Library, Vol. 17, pp. 195-206, 1998.
- [10] F. Lamarche, *Topoplan : a topological path planner for real time human navigation under floor and ceiling constraints*, Computer Graphics Forum, Wiley Online Library, vol. 28, pp. 649-658, 2009.
- [11] S. Beckhaus, Steffi, F. Ritter and T. Strothotte, *Cubical path-dynamic potential fields for guided exploration in virtual environments*, Conference on Computer Graphics and Applications, IEEE, pp. 387-459, 2000.



FIGURE 9 – Navigation automatique en extérieur.



FIGURE 10 – Navigation automatique en intérieur.

- [12] N. Halper, R. Helbing and T. Strothotte, *A Camera Engine for Computer Games : Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence*, Computer Graphics Forum, Wiley Online Library, vol. 20, pp. 174-183, 2001.
- [13] W. Bares, S. McDermott, C. Boudreaux, S. Thainimit, *Virtual 3D camera composition from frame constraints*, ACM international conference on Multimedia, pp. 177-186, 2000.
- [14] N. Courty and E. Marchand, *Computer animation : A new application for image-based visual servoing*, Conference on Robotics and Automation, ICRA, IEEE, vol. 1, pp. 223-228, 2001.
- [15] B. Tomlinson, B. Blumberg and D. Nain, *Expressive autonomous cinematography for interactive virtual environments*, International conference on Autonomous agents, ACM, pp. 317-324, 2000.
- [16] O. Bourne, A. Sattar and S. Goodwin, *A constraint-based autonomous 3d camera system* Constraints, Springer, Vol. 13, pp. 180-205, 2008.
- [17] W. Bares, L. Zettlemoyer, D. Rodriguez and J. Lester, *Task-sensitive cinematography interfaces for interactive 3D learning environments*, International conference on Intelligent user interfaces, ICM, pp. 81-88, 1998.
- [18] J. Feddema and O. Mitchell, *Vision-guided servoing with feature-based trajectory generation for robots*, IEEE Transactions on Robotics and Automation, Vol. 5, pp. 691-700, 1989.
- [19] E. Marchand and G. Hager, *Dynamic sensor planning in visual servoing*, IEEE International Conference on Robotics and Automation, vol. 3, pp. 1988-1993, 1998.
- [20] G. Hager, *A modular system for robust positioning using feedback from stereo vision*, IEEE Transactions on Robotics and Automation, vol. 3, pp. 582-595, 1997.
- [21] A. Dame and E. Marchand, *Entropy-based visual servoing*, International Conference on Robotics and Automation, ICRA, IEEE, pp. 707-713, 2009.
- [22] F. Maes, Frederik, A. Collignon, D. Vandermeulen, G. Marchal and P. Suetens, *Multimodality image registration by maximization of mutual information*, IEEE Transactions on Medical Imaging, vol. 16, pp. 187-198, 1997.
- [23] J. Lapresté and Y. Mezouar *A Hessian approach to visual servoing*, Intelligent Robots and Systems IROS, IEEE, vol. 1, pp. 998-1003, 2004.
- [24] M. Ozaki, L. Gobeawan, S. Kitaoka, H. Hamazaki, Y. Kitamura and R. Lindeman, *Camera movement for chasing a subject with unknown behavior based on real-time viewpoint goodness evaluation*, The Visual Computer, Springer, vol. 26, pp. 629-638, 2010.
- [25] C. Ware and S. Osborne, *Exploration and virtual camera control in virtual three dimensional environments*, ACM SIGGRAPH Computer Graphics, vol. 24, pp. 175-183, 1990.
- [26] <http://www.ogre3d.org/>.