



Un modèle probabiliste pour la reconstruction 3D

Adrien Chan-Hon-Tong

► **To cite this version:**

Adrien Chan-Hon-Tong. Un modèle probabiliste pour la reconstruction 3D. RFIA 2012 (Reconnaissance des Formes et Intelligence Artificielle), Jan 2012, Lyon, France. pp.978-2-9539515-2-3, 2012. <hal-00660979>

HAL Id: hal-00660979

<https://hal.archives-ouvertes.fr/hal-00660979>

Submitted on 19 Jan 2012

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.

Un modèle probabiliste pour la reconstruction 3D

Adrien CHAN-HON-TONG

CEA, LIST, Laboratoire Vision et Ingénierie des Contenus

France 91191 Gif Sur Yvette

adrien.chan-hon-tong@cea.fr

Résumé

Cet article introduit un cadre probabiliste permettant d'étudier mathématiquement et a priori des algorithmes de reconstruction 3D. Dans cet article, l'étude se concentre sur la famille des algorithmes de reconstruction 3D définis par les trois critères suivants. Premièrement, ils étudient une scène observée simultanément par deux caméras calibrées. Deuxièmement, ils discrétisent la scène en un ensemble de voxels pleins ou vides. Troisièmement, ces algorithmes se basent sur la donnée d'une fonction qui à chaque voxel associe une mesure de la concordance entre les zones des deux images où ce voxel se projette.

L'idée-clé de ce cadre probabiliste est que, statistiquement, la concordance entre les zones des deux images où se projette un voxel est : forte si ce voxel est sur la surface que l'on cherche à reconstruire, et, faible sinon. Nous montrons d'abord que cette idée-clé, assez naturelle, est justifiée expérimentalement.

Nous montrerons alors que certaines propriétés de ces algorithmes peuvent se déduire mathématiquement des deux distributions de probabilité des concordances (celle pour les voxels sur la surface et celle pour ceux en dehors de la surface). Nous montrerons pour quelques algorithmes que les propriétés établies théoriquement à l'aide de notre modélisation sont compatibles avec les propriétés expérimentales mesurées sur des jeux de données réelles. Enfin, nous étudierons théoriquement une reconstruction NP-complète, que l'on ne peut pas étudier expérimentalement.

Mots Clef

modélisation, probabilité, mathématiques, reconstruction 3D.

Abstract

This article offers a probabilistic model that allows us to evaluate mathematically accuracy of some 3D reconstruction algorithms without any experiment. This article is focused on the family of 3D reconstruction algorithms defined by the following three points. First, they study a scene observed simultaneously by two calibrated cameras. Then, they represent the scene as a set of voxel. Finally, they take as input a function which gives for every voxel, a measure of the concordance between the areas in the two pictures where the voxel project himself.

The main idea of this probabilistic model is the fact that statistically the concordance between the two areas in the two pictures where the voxel project himself is : strong if the voxel is on the observed surface, and, weak otherwise. First, we show that this idea is validated by experiments.

Then, we show that for algorithms of the introduced family, accuracy can be compute by using only the two probability density functions associated to the concordances (one for the voxel on the surface and one for the voxel outside the surface). We show that the predicted accuracy is compatible with the experimental accuracy on real data. Finally, we study mathematically an NP-complete reconstruction, which cannot be studied experimentally.

Keywords

probability, modelisation, mathematics, 3D reconstruction

1 Introduction

Le problème de la reconstruction 3D consiste à déduire la géométrie d'un objet à partir d'un ensemble de mesures. Sur le plan mathématique, on le réduit le plus souvent au problème d'approximation d'une surface à partir d'un échantillonnage de points, associé à des hypothèses de régularité sur cette surface. Sur le plan algorithmique, il consiste la plupart du temps, à réaliser, à partir de mesures laser et/ou vidéo, un maillage, ou un ensemble de voxels, représentant la géométrie d'un objet. La reconstruction 3D aboutit alors à une numérisation de l'objet étudié.

Ce problème de numérisation d'objets 3D est motivé par un grand nombre d'applications possibles. On peut citer le contrôle de qualité, la robotique, la réalité virtuelle, la réalité augmentée, la reconnaissance de formes... Récemment, les performances des algorithmes de reconstruction 3D ont connu des améliorations majeures. De plus, suite à l'apparition de méthodes moyennement fiables mais très faciles à mettre en oeuvre, la reconstruction 3D commencent aujourd'hui à toucher le grand public ; alors qu'elle était autrefois principalement présente dans l'industrie via des méthodes de reconstruction par laser, et dans le monde académique.

Cependant, aujourd'hui, la qualité des méthodes n'est évaluée qu'en comparant la surface numérisée à une vérité terrain. En effet, pour la majorité des méthodes de numérisation, on ne dispose pas d'une estimation a priori de leur

efficacité.

Aussi, dans cet article, nous nous intéressons à un cadre probabiliste du problème de la reconstruction 3D permettant de calculer mathématiquement, a priori, l'efficacité des algorithmes de reconstruction 3D. L'idée clé de cette modélisation est de voir l'entrée de tout algorithme de reconstruction, par exemple les images de la scène et les informations contextuelles, comme étant le résultat d'un tirage probabiliste. Or, si l'entrée de l'algorithme est un tirage probabiliste, alors il en est de même pour la sortie. Ainsi, statistiquement, la précision du résultat d'un algorithme de reconstruction 3D ne dépend que de la fonction de répartition associée à son entrée.

De façon plus précise, nous nous concentrerons sur la famille des algorithmes de reconstruction 3D qui vérifient les points suivants. Ils étudient une scène observée simultanément par deux caméras calibrées. Ils discrétisent la scène en un ensemble de voxel plein ou vide. Ils se basent sur la donnée, pour chaque voxel, d'une mesure de la concordance entre les deux zones des deux images où ce voxel se projette. L'intérêt de cette famille est que, d'une part, la distribution des mesures de concordance possède de forts invariants statistiques et que, d'autre part, la discrétisation en voxel permet de se ramener à un nombre fini de mesures. L'article est structuré comme suit : la prochaine section est consacrée à l'état de l'art des formulations mathématiques de la reconstruction 3D et des méthodes de résolution de ce problème. La section 3 introduit plus précisément la famille des algorithmes étudiée. La section 4 donne deux exemples explicites de mesure de concordance et montre que pour chacune de ces mesures, les distributions associées exhibent des invariants statistiques. Dans la section 5, nous donnons une définition mathématique de l'erreur statistique associée à un algorithme de reconstruction 3D, et nous montrons pour deux algorithmes comment calculer explicitement l'erreur théorique que nous comparons ensuite avec l'erreur expérimentale sur des jeux de données réelles. Dans la section 6, nous utilisons notre modèle pour étudier théoriquement une reconstruction donnée NP-complète. La section 7 conclut l'article et ouvre des perspectives de travail futur.

2 L'état de l'art en reconstruction 3D

Dans l'état de l'art, de nombreux algorithmes proposent des solutions au problème de la reconstruction 3D. Nous nous focaliserons ici sur les méthodes travaillant à partir d'images obtenues à l'aide de caméras calibrées (ie la relation géométrique entre l'espace 3D et le repère image de la caméra est connue).

Cette famille d'approches comporte deux catégories : les méthodes passives et les méthodes actives. Dans les méthodes actives, l'acquisition d'images est effectuée avec un éclairage structurant [1]. Cet éclairage permet d'assurer la présence de points de contrôle à la surface de l'objet, ce qui facilite l'étape suivante de reconstruction.

Dans les méthodes passives, on effectue une acquisition

d'images sous un éclairage non contrôlé. Les méthodes passives sont elles-mêmes divisibles en de nombreuses sous-familles. Parmi ces sous-familles, on trouve les méthodes qui définissent la surface par le résultat d'une minimisation notamment par la technique dite des coupes de graphe [2, 3, 4]. La théorie des coupes de graphe a été développée dans [5, 6]. D'autres méthodes cherchent à reconstruire la géométrie à partir d'éléments connus : par déformation de modèles [7] par exemple, ou par assemblage de primitives 3D simples, telles que des disques [8] ou des rectangles [9]. Les méthodes volumétriques reposent sur une idée encore différente : il s'agit de produire une surface sous la forme d'un ensemble de voxels. Ces algorithmes ont en particulier connus un essor après [10] où S. M. Seitz et al. ont introduit la méthode de coloration des voxels (voxel coloring). Plus tard, la méthode de sculpture d'espace (Space Carving) [11] a proposé une approche itérative pour affiner photo par photo une surface initiale. Dans [12], une approche demandant moins de ressource machine, basée sur la hiérarchisation des voxels, est proposée. Certains algorithmes sont le résultat de combinaisons de plusieurs types de méthodes et proposent une chaîne de traitement d'images. Par exemple, dans [13] V.H. Hiep et al proposent de reconstruire la géométrie d'un objet en trois étapes. Tout d'abord, un nuage de points dense est obtenu par couplage de points d'intérêt. Puis, une première surface est créée à partir du nuage de points par minimisation d'un critère (coupes de graphe) combinant une mesure de photoconsistance et l'erreur de visibilité. Enfin, cette première surface est raffinée par un processus d'optimisation variationnel local. Cette dernière méthode illustre l'état de l'art de la reconstruction 3D : l'algorithme est efficace et produit des objets de grande qualité, mais il reste heuristique.

Il existe aussi des méthodes non heuristiques, basées sur une modélisation mathématique. La première modélisation mathématique de la reconstruction 3D est la théorie de l'approximation d'une surface par un échantillonnage fini. La méthode du Crust [14] fut la première méthode justifiant le bien fondé de la reconstruction 3D : le Delaunay restreint de l'échantillon produit par cette méthode, est capable (après post-traitement) d'approximer précisément la surface sous-jacente si cette surface est suffisamment régulière. Cependant, cette méthode suppose que l'échantillonnage de la surface soit peu bruité. Cela pose un problème pour les nuages de points 3D provenant de mesures laser ou a fortiori d'images. Dans [15], F. Chazal propose de modéliser les surfaces par des mesures pour être capable, grâce à la théorie de la mesure, de donner un cadre formel à la reconstruction de surfaces à partir d'un nuage de points bruité. Mais, la reconstruction 3D ne peut se résumer à l'approximation de surface par un échantillonnage. En effet, l'information contenue dans n'importe quelle vidéo est plus riche que celle d'un simple nuage de point. Une autre approche est de chercher à distinguer les zones visibles de celles qui ne le sont pas. De nombreux algorithmes de l'état de l'art se basent sur une modélisation de

cette contrainte de visibilité comme dans [10] (Ordinal Visibility Constraint). Enfin, une approche qui nous intéresse particulièrement consiste à voir le problème d'un point de vue statistique. Par exemple, R. Bhotika et al. [17] forment une théorie qui introduit la probabilité qu'un voxel soit plein ou vide.

Cette approche statistique qu'on retrouve aussi dans [18] permet de concevoir des algorithmes. Mais on peut aller encore plus loin : dans cet article nous allons montrer que cette approche statistique pourrait permettre d'évaluer, a priori, la qualité d'une reconstruction produite par un algorithme.

3 La famille des algorithmes étudiés

Dans cet article, nous nous concentrons sur l'ensemble des algorithmes vérifiant les propriétés suivantes :

- Ils étudient une scène observée simultanément par deux caméras calibrées.
- Ils supposent que l'espace est constitué de voxels qui sont soit parfaitement pleins et opaques, soit parfaitement vides et transparents.
- Ils cherchent à déterminer l'ensemble des voxels à la fois pleins et visibles (ce qui correspond globalement aux voxels sur la surface observée).
- Ils se basent sur la donnée, pour chaque voxel, d'une mesure de la concordance entre les zones des deux images où se projette le voxel.

Formellement, on considère I un ensemble fini de voxels dans le champ de visibilité joint des deux caméras. Pour tout voxel n dans I , on note $L(n)$ le sous-ensemble, supposé connu, des voxels de I qui peuvent cacher n à au moins une des deux caméras. Par exemple, si les deux caméras sont de type « pinhole », $L(n)$ sera l'ensemble des voxels intersectant l'union des deux segments reliant le voxel n aux centres optiques des deux caméras (plus généralement, $L(n)$ est facilement calculable à partir de la calibration des deux caméras). Cela est illustré dans la figure 1. De même, $L^*(n)$ sera l'ensemble des voxels cachant ou pouvant être caché par le voxel n .

Pour les algorithmes étudiés, une géométrie x pour la scène observée est naturellement un élément de $\{0, 1\}^I$. $x(n) = 1$ signifiant que le voxel n est plein, $x(n) = 0$ signifiant que le voxel n est vide. De même, la donnée, pour chaque voxel n , d'une mesure de la concordance $y(n)$ entre les zones dans les deux images où se projette le voxel n , est naturellement un élément de $[0, 1]^I$. $y(n) = 1$ correspond à une concordance parfaite, $y(n) = 0$ correspondant à une concordance nulle.

Un algorithme de la famille étudiée n'est alors rien d'autre qu'une fonction qui à un élément de $[0, 1]^I$ fait correspondre un élément de $\{0, 1\}^I$.

Remarquons qu'une géométrie sous la forme d'un étiquetage plein/vidé des voxels est bien une réponse au problème de la reconstruction 3D. En effet, cela permet de déterminer les voxels visibles depuis les deux caméras et ceux qui ne

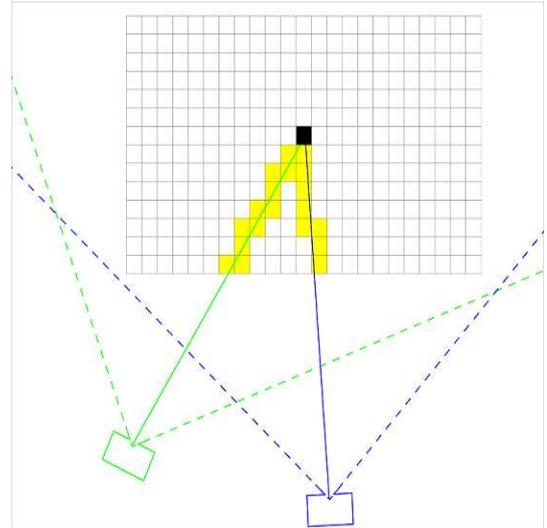


FIGURE 1 – Schéma de la situation.

Chaque case de la grille représente un voxel. Les demi-cônes vert et bleu représenté en pointillé représentent les champs de visibilité des deux caméras. Ici, les deux caméras sont de type « pinhole » ainsi le voxel noir n peut être caché à une des deux caméras par chacun des voxels jaunes constituant $L(n)$.

le sont pas car on suppose que les voxels pleins sont totalement opaques et que les voxels vides sont totalement transparents. De même, cela permet de déterminer l'ensemble, qu'on note $P(x)$, des voxels à la fois pleins et visibles dans les deux images, c'est à dire, globalement, les voxels de la surface. Cela permet donc aussi de déterminer l'ensemble des autres voxels qu'on note $N(x)$.

L'idée-clé de cette article est que les algorithmes de la famille étudiée ont une sortie entièrement déterminée par la donnée de la concordance, elle-même vue comme le résultat d'un tirage aléatoire selon une certaine probabilité.

En effet, un algorithme de reconstruction 3D fonctionne sur la supposition implicite que la concordance va être statistiquement forte pour les voxels pleins et visibles et faible pour les autres. Cet invariant statistique permet de donner un sens, pour les algorithmes de la famille étudiée, à la probabilité que l'algorithme détermine la surface exacte. En effet, le fait qu'un algorithme détermine la bonne surface n'est qu'un événement de notre univers auquel on peut donc associer une probabilité. Ainsi, cet invariant statistique nous permet d'étudier mathématiquement ces algorithmes.

Pour donner un sens mathématique à cela, notons $\Omega = \{0, 1\}^I \times [0, 1]^I$. Ω est naturellement l'univers sur lequel il convient de définir une probabilité \mathbb{P} .

L'observation d'une scène avec les deux caméras se modélise donc comme une tirage aléatoire de $(x, y) \in \Omega$ selon une probabilité \mathbb{P} qui prend en compte l'invariant statistique associé à la concordance (plus forte pour les voxels

de la surface que pour les autres) ainsi que la texturation de la scène (plus une scène est texturée, moins les valeurs des concordances seront concentrées autour de 0.5).

Ainsi, la probabilité qu'un algorithme *rec* détermine la bonne surface est $\mathbb{P}(\{(x, y) / \text{rec}(y) = x\})$. De façon plus générale, on peut étudier les propriétés de ces algorithmes dans notre modélisation probabiliste. Par exemple, la probabilité que l'erreur en norme L_1 entre la surface réel et la surface reconstruite soit exactement k est aussi un événement intéressant associé à la probabilité $\mathbb{P}(\{(x, y) / \|\text{rec}(y) - x\|_1 = k\})$

Mais avant l'étude de ces algorithmes dans ce cadre, il convient de démontrer expérimentalement que la concordance vérifie bien cet invariance statistique. Pour cela, il convient de donner des exemples concrets de mesure de concordances.

4 Mesure de concordances

Dans cette partie, nous allons donner deux exemples d'implémentations possibles pour la concordance et montrer que pour ces deux implémentations, on vérifie que la concordance est : statistiquement forte pour les voxels pleins et visibles, et, faible pour les autres.

Remarquons d'abord que cette supposition est assez naturelle.

Si un bout de matière coloré ie un voxel plein est observé (donc visible) par les deux caméras, les pixels des zones de projection du voxel dans les deux images auront statistiquement la couleur du voxel. Donc, la concordance entre ces deux zones sera forte.

Réciproquement, deux voxels différents (et pleins) ont statistiquement des couleurs différentes (ceci est d'autant plus juste que la scène est texturée). Ainsi, si un voxel est vide ou caché, alors les couleurs des pixels dans les deux images dans les zones de projection de ce voxel, sont en fait les couleurs de deux autres voxels (les premiers pleins sur la route des rayons entre les centres optiques et le voxel). Donc, statistiquement, ces couleurs sont différentes et la concordance associée au voxel est faible.

4.1 Correlation croisée normalisée

La corrélation croisée (CC) est un outil standard pour mesure la correspondance entre deux signaux. Soit f et g deux fonctions définies sur un domaine K à valeur dans un espace vectoriel muni d'un produit scalaire, alors la corrélation croisée $CC_{f,g,K}$ est définie comme

$$\int_K \left(f(u) - \frac{1}{|K|} \int_K f(s) ds \right) \left(g(u) - \frac{1}{|K|} \int_K g(s) ds \right) du$$

(ici $|\cdot|$ représente la mesure de l'ensemble)

La corrélation croisée normalisée (NCC) se définit alors comme :

$$NCC_{f,g,K} = \frac{CC_{f,g,K}}{\sqrt{CC_{f,f,K}CC_{g,g,K}}}$$

Dans notre cas présent, notons C_1 la fonction qui à un point de l'espace 3D contenu dans le champs de visibilité joint des deux caméras associe la couleur du pixel correspondant dans l'image 1 (le pixel dans lequel est projeté le point). Notons, de même, C_2 pour la couleur du pixel correspondant dans l'image 2. La corrélation croisée normalisée au niveau d'un voxel n est $NCC(n) = NCC_{C_1,C_2,n}$

Une implémentation possible de la concordance est alors $y(n) = 0.5 + 0.5 \times NCC(n)$ (pour que la concordance ait une valeur entre 0 et 1). On prend $y(n) = 0.5$ dans le cas où $CC_{C_1,C_1,n} = CC_{C_1,C_1,n} = CC_{C_1,C_2,n} = 0$ (texturation nulle et couleur identique), et, $y(n) = 0$ dans le cas où $CC_{C_1,C_1,n} = 0$ ou $CC_{C_1,C_1,n} = 0$.

4.2 Erreur de projections croisées

Nous introduisons ici une mesure d'erreur non standard que nous nommerons l'erreur de projections croisées (PC) au niveau d'un voxel n . Elle est définie ainsi :

$$PC(n) = \int_n \|C_1(p) - C_2(p)\|_2^2 dp$$

Intuitivement, plus cette erreur est grande plus les couleurs des pixels des zones de projection du voxel sont différentes entre les deux images.

L'erreur de projections croisées normalisée (PCN) se définit alors comme

$$PCN(n) = \frac{PC(n)}{\max\{PC(m) / m \in L^*(n)\} + \delta}$$

où $\delta \ll 1$ est une constante positive.

Une implémentation possible de la concordance est alors $y(n) = 1 - PCN(n)$

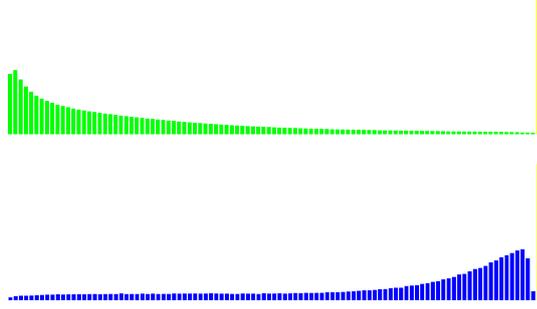
4.3 Distribution des valeurs des concordances

Nous avons calculé les concordances des voxels pour les deux implémentations introduites (basées sur la NCC ou la PCN) sur des jeux de données extraits de [19].

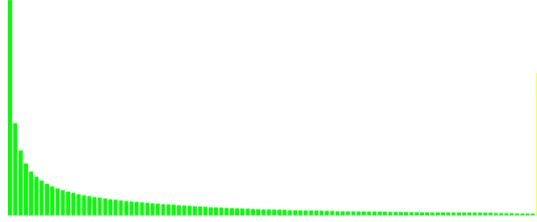
Le protocole expérimental est le suivant. On choisit un modèle 3D et de deux images associées. Les calibrations des caméras, les boîtes englobantes sont celles fournies avec les images et le modèle. On forme la vérité terrain sous-forme de voxel à l'aide de la vérité terrain sous la forme d'un maillage en calculant l'ensemble des voxels intersectés par le maillage (les voxels sont des cubes de 0.05 de côté). Les voxels intersectés par le maillage sont les voxels pleins (pleins seulement, pas nécessairement visibles). On calcule alors pour chaque voxel n , $y(n)$. On forme l'histogramme de la distribution des valeurs de concordance pour les voxels de $P(x)$ (pleins et visibles), et, de même pour les autres voxels (ceux de $N(x)$). Les histogrammes sont formées de 100 bandes. (Ainsi, dans l'histogramme associé aux voxels de $P(x)$, la hauteur de la bande i est $\frac{1}{|P(x)|} |\{n \in P(x) / y(n) \in [\frac{i}{100}, \frac{i+1}{100}]\}|$)

Ces histogrammes (figure 2) confirme que la concordance est statistiquement forte pour les voxels pleins et visibles et

FIGURE 2 – Distribution des valeurs de concordance. Le vert correspond aux voxels hors de la surface. Le bleu correspond aux voxels dans la surface. La barre jaune indique 0.25 pour donner l'échelle.



Ici, le modèle utilisé est fountain, les images utilisées sont les images 4 et 6, la concordance est basée sur la NCC.



Ici, le modèle utilisé est herzjesu, les images utilisées sont les images 4 et 3, la concordance est basée sur la NCC.

faible pour les autres. Ce qui justifie notre approche statistique.

5 Etude statistique d'algorithmes de reconstruction 3D

Dans cette section, nous précisons d'abord notre modélisation probabiliste. Puis, nous introduisons trois algorithmes de reconstruction. Enfin, nous étudions, via la modélisation, deux algorithmes et nous comparons les résultats théoriques aux résultats expérimentaux.

5.1 La modélisation probabiliste retenue

\mathbb{P} la densité probabilité dont on munit Ω doit vérifier un certain nombre de conditions :

- En l'absence de connaissances a priori sur la scène observée, toutes les scènes doivent être équiprobables.
- Il doit être probable que $y(n)$ soit proche de 1 si n est dans $P(x)$ et proche de 0 sinon ie dans $N(x)$.
- La forme exacte de la distribution doit pouvoir dépendre de la texturation de la scène.

De plus, il est intéressant que la densité de probabilité \mathbb{P} vérifie :

- La probabilité qu'un voxel n soit plein sachant y et sachant qu'il n'est pas visible doit être 0.5, c'est à dire,

qu'on n'a aucune information sur les voxels non visibles. Formellement, cela correspond à :

$$\mathbb{P}(X_n = 1 | Y = y, \neg(\forall m \in L(n) X_m = 0)) = 0.5$$

(X_n , et Y sont les variables aléatoires canoniques associées à $x(n)$ et y , \neg est le symbole de la négation logique)

- $y(n)$ doit être lié voire égal à la probabilité que le voxel n soit plein sachant y et sachant qu'il est visible. Cela correspond à :

$$\mathbb{P}(X_n = 1 | Y = y, \forall m \in L(n) X_m = 0) = y(n)$$

Pour répondre à ces contraintes, la forme choisie pour la densité de probabilité \mathbb{P} est : $\forall (x, y) \in \Omega$

$$\mathbb{P}(x, y) = 2^{-|I|} \times$$

$$\left(\prod_{n \in P(x)} y(n) C(y(n)) \right) \left(\prod_{n \in N(x)} (1 - y(n)) C(y(n)) \right)$$

où $u \rightarrow C(u)$ est une fonction positive qui dépend de la texturation de la scène observée (et qui est normalisée pour que \mathbb{P} soit une probabilité).

Observer une scène avec les deux caméras correspond donc à effectuer un tirage aléatoire de $(x, y) \in \Omega$ selon la probabilité \mathbb{P} , et, reconstruire cette scène à l'aide d'un algorithme *rec* revient à calculer *rec*(y).

5.2 Quelques algorithmes

Algorithme naïf. L'algorithme naïf, que l'on notera *rec*₀ par la suite, consiste à décider qu'un voxel est plein dès que sa probabilité d'être plein sachant y est supérieur à sa probabilité d'être vide sachant y ie $\mathbb{P}(X_n = 1 | Y = y) > \mathbb{P}(X_n = 0 | Y = y)$. Or cela équivaut à $y(n) > 0.5$, puisqu'en utilisant la formule des probabilités totales on établit l'égalité :

$$\begin{aligned} & \mathbb{P}(X_n = 1 | Y = y) \\ &= \mathbb{P}(X_n = 1 | Y = y, \forall m \in L(n) X_m = 0) \\ & \quad \times \mathbb{P}(\forall m \in L(n) X_m = 0 | Y = y) \\ &+ \mathbb{P}(X_n = 1 | Y = y, \neg(\forall m \in L(n) X_m = 0)) \\ & \quad \times \mathbb{P}(\neg(\forall m \in L(n) X_m = 0) | Y = y) \\ &= 0.5 + (y(n) - 0.5) \prod_{m \in L(n)} \mathbb{P}(X_m = 0 | Y = y) \end{aligned}$$

Ainsi, on a $(rec_0(y))(n) = 1 \Leftrightarrow y(n) > 0.5$

Algorithme glouton. L'algorithme glouton, que l'on notera *rec*₁ par la suite, consiste à trier les voxels par ordre de $y(n)$ décroissant, puis de parcourir selon cet ordre les voxels, et de décider que le voxel n est plein s'il ne cache aucun des voxels décidés comme plein et si $y(n) > 0.5$.

Algorithme argmax. L'algorithme argmax, que l'on notera rec_∞ par la suite, consiste à déterminer la géométrie la plus probable sachant y . En ce sens cet algorithme est un processus d'inférence bayésienne.

Formellement, on définit :

$$\begin{aligned} rec_\infty(y) &= \underset{x \in \{0,1\}^I}{\operatorname{argmax}} (\mathbb{P}(X = x | Y = y)) \\ &= \underset{x \in \{0,1\}^I}{\operatorname{argmax}} (\log(\mathbb{P}(x, y))) \end{aligned}$$

(X est la variable aléatoire canonique associée à x)

Nous montrons ici que le problème du calcul de la surface la plus probable peut se ramener à un problème d'optimisation combinatoire classique.

Observons que pour toute géométrie x , pour tout voxel n , et pour tout voxel m dans $L^*(n)$, il est impossible que simultanément n et m soient dans $P(x)$ (c'est à dire plein et visibles à la fois dans les deux images). En effet, devant un voxel visible, les voxels sont vides ; et derrière un voxel plein, les voxels sont occultés. On notera $W =$

$$\left\{ w \in \{0,1\}^I / \forall n \in I \forall m \in L^*(n) \ x(n) + x(m) \leq 1 \right\}$$

On peut ainsi associer à une géométrie un élément de W . Réciproquement, soit $w \in W$, notons x_w la géométrie tel que $\forall n \in I, x_w(n) = w(n)$. Il est immédiat que $P(x_w) = \{n \in I / w(n) = 1\}$ et $N(x_w) = \{n \in I / w(n) = 0\}$. Cette correspondance entre vecteurs soumis à une contrainte algébrique W et géométries, est une bijection si on ne considère que les géométries où l'ensemble des voxels non visibles sont vides. Or, du point de vue de \mathbb{P} toute géométrie est équivalente à la géométrie, identique sur l'ensemble des voxels visibles, et où l'ensemble des voxels non visibles sont vides.

Ainsi, on établit les égalités suivantes :

$$\begin{aligned} &\underset{x \in \{0,1\}^I}{\operatorname{argmax}} (\log(\mathbb{P}(x, y))) \\ &= \underset{w \in W}{\operatorname{argmax}} \left(\sum_{n \in I} (\log(y(n)) - \log(1 - y(n))) w(n) \right) \end{aligned}$$

Or, cette dernière équation est un problème d'optimisation combinatoire classique de type programme linéaire en nombre entier (PLNE) car W est formé de contraintes linéaires sur des variables entières, et que le critère à optimiser est linéaire. Ce PLNE est, dans cet article, donné comme NP-complet (il est un cas particulier, mais néanmoins assez général, du problème NP-complet couramment appelé « stable »).

Remarquons que le PLNE correspondant à la surface la plus probable semble similaire au PLNE suivant :

$$\underset{w \in W}{\min} \left(\sum_{n \in I} PC(n) w(n) \right)$$

où PC est l'erreur de projections croisées. Ce dernier semble permettre de construire la surface qui minimise la somme des erreurs de projection. En effet, $w(n) = 1$ représenterait le fait que le voxel n soit plein et visible. Ainsi, on minimiserait la somme des erreurs de projection croisée au niveau des voxels pleins et visibles. Néanmoins, la solution de ce dernier PLNE est la solution triviale (tous les voxels vides). Intuitivement il ne faut pas seulement pénaliser les erreurs, il faut aussi encourager le fait qu'un voxel visible soit plein quand les deux zones dans les deux images où il se projette sont très concordantes. Ce qui est fait dans le calcul de la surface la plus probable.

Il convient maintenant, afin de valider notre modélisation, de montrer que pour des algorithmes que l'on peut exécuter, on obtient des résultats concordants entre les propriétés théoriques et expérimentales.

5.3 Calcul a priori et mesure d'erreur

Calcul de la probabilité de succès pour l'algorithme naïf. L'algorithme naïf ne fait aucune erreur si, pour tout voxel n , $y(n)$ est inférieur à 0.5 si n est dans $N(x)$ et $y(n)$ est supérieur à 0.5 si n est dans $P(x)$. On établit donc que la probabilité de succès de l'algorithme naïf ps_0 est égale à

$$\left(\int_{0.5}^1 u C(u) du \right)^{|P(x)|} \left(\int_0^{0.5} (1-u) C(u) du \right)^{|N(x)|}$$

Calcul de la probabilité de succès pour l'algorithme glouton. L'algorithme glouton ne fait aucune erreur si, pour tout voxel $n \in P(x)$ $y(n) \geq 0.5$ et si n est celui qui a la plus grande concordance parmi les voxels de $L^*(n)$. Ainsi on établit l'inégalité :

$$ps_1 \gtrsim \left(\int_{0.5}^1 u C(u) \left(\int_0^u (1-v) C(v) dv \right)^{|L^*(n)|} du \right)^{|P(x)|}$$

En effet, le terme de droite est construit comme si les ensembles $L^*(n)$ étaient disjoints, et, de ce fait, certaines intégrales de la forme $\int_0^u (1-v) C(v) dv$ apparaissent plus de fois qu'il ne le faudrait pour ps_1 . Mais comme ces intégrales sont des probabilités, on sait qu'elles sont inférieures à 1, ce qui permet d'établir l'inégalité. De plus, ces intégrales sont proches de 1, ce qui nous autorise à faire l'approximation.

Globalement, par rapport à ps_0 , on a remplacé $\int_0^{0.5} (1-v) C(v) dv$ par $\int_0^u (1-v) C(v) dv$ où $u \geq 0.5$ et est statistiquement proche de 1.

Comparaison du calcul théorique et de la mesure. Nous avons rapproché des résultats théoriques fournis par notre modélisation et des résultats expérimentaux obtenus sur des jeux de données extraits de [19].

Le protocole expérimental est le suivant. D'abord, on calcule les concordances comme dans l'expérience précédente. On calcule à la suite une fonction de concentration, ie C dans la modélisation, en utilisant la vérité terrain. On choisit un algorithme. Pour cette fonction de concentration C et cet algorithme rec , on calcule (de façon approchée) l'erreur moyenne ie $\sum_{k \in \mathbb{N}} k \mathbb{P}(\|rec(Y) - X\|_1 = k)$. Indépendamment on calcule la reconstruction produite par l'algorithme ie $rec(y)$ et on mesure l'erreur avec la vérité terrain.

Le tableau 1 résume le résultat des expériences.

jeu de données	expérience	calcul théorique
fountain 46 NCC	0.46	0.44
herzjesu 43 PCN	0.49	0.48
Résultat pour l'algorithme naïf.		
jeu de données	expérience	calcul théorique
fountain 46 NCC	0.03	0.06
herzjesu 43 PCN	0.01	0.02

Résultat pour l'algorithme glouton.

La case jeu de données indique : vérité terrain, photos utilisées, méthode de calcul de la concordance.

TABLE 1 – Comparaison de l'erreur expérimentale et de l'erreur moyenne théorique en proportion du nombre de voxels.

On observe que l'erreur mesurée est cohérente avec l'erreur moyenne théorique. Cela justifie notre approche statistique.

Remarquons que les résultats théoriques énoncés sur les algorithmes ne dépendent que de la fonction de concentration (C dans notre modélisation) et non de la vérité terrain elle-même. Ici, la concentration est calculée à partir de la vérité terrain. Mais, il n'est pas impossible qu'en améliorant notre modélisation, on puisse déterminer C en utilisant uniquement la distribution des valeurs de concordance (sans la vérité terrain ie sans savoir quels sont les voxels pleins et visibles). Par exemple, si $\forall u \in [0, 1] C(u) = \lambda(\sigma) \exp\left(-\frac{(u-0.5)^2}{\sigma^2}\right)$ avec λ le coefficient de normalisation (pour avoir des probabilités) et σ un paramètre qui dépend de la texturation, il est possible de déterminer σ , sans utiliser la vérité-terrain, en calculant $\frac{1}{|I|} \sum_{n \in I} (y(n) - 0.5)^2$.

De façon plus générale, si C est une mixture de gaussienne de taille fixe, symétrique par rapport à 0.5, comme on sait calculer $\left\{ \int_0^1 u(u-0.5)^{2k} C(u) du \mid k \in \mathbb{N} \right\}$ à partir de $\{y(n) \mid n \in I\}$, et que $|P(x)| \leq |N(x)|$, il est possible de pouvoir retrouver C sans utiliser la vérité terrain.

Etudier les algorithmes relativement à une fonction de concentration est un premier pas vers la possibilité de donner aux algorithmes des garanties statistiques ne dépendant que de la distribution des valeurs de concordance.

Cela nous invite à étudier l'algorithme basé sur le calcul de la surface la plus probable qui ne peut pas en pratique être

testé du fait de sa complexité.

6 Probabilité que la surface la plus probable soit la surface réelle

6.1 Lemme

$\forall p$ une densité de probabilité sur $]0, 1[$ dont on note $P(u) = \int_u^1 p(s) ds$.

On tire u_1, \dots, u_r selon p dans $]0, 1[$ et on note A_r l'événement

$$\prod_{i \in \{1, \dots, r\}} u_i \geq \prod_{i \in \{1, \dots, r\}} (1 - u_i)$$

Alors $PLemme_r$ la probabilité de l'événement A_r vaut :

$$\int_{[0,1]^{r-1}} P \left(\frac{\prod_{i \in \{2, \dots, r\}} \frac{1-u_i}{u_i}}{1 + \prod_{i \in \{2, \dots, r\}} \frac{1-u_i}{u_i}} \right) \prod_{i \in \{2, \dots, r\}} p(u_i) du_2 \dots du_r$$

en effet A_r est équivalent à l'événement

$$u_1 \geq \frac{\prod_{i \in \{2, \dots, r\}} \frac{1-u_i}{u_i}}{1 + \prod_{i \in \{2, \dots, r\}} \frac{1-u_i}{u_i}}$$

6.2 Surface la plus probable

Dans cet article, nous ne donnons pas de formule pour ps_∞ la probabilité que la surface la plus probable soit la surface réelle. Néanmoins dans où C est symétrique par rapport à 0.5, on peut faire les observations suivantes :

$$\begin{aligned} ps_\infty &= \mathbb{P}(\{(x, y) / rec_\infty(y) = x\}) \\ &= \mathbb{P}\left(\{(x, y) / \forall z \in \{0, 1\}^I \mathbb{P}(x, y) \geq \mathbb{P}(z, y)\}\right) \end{aligned}$$

Or

$$\begin{aligned} &\mathbb{P}(x, y) \geq \mathbb{P}(z, y) \\ \Leftrightarrow &\left(\prod_{n \in (P(x) - P(z))} y(n) \right) \left(\prod_{n \in (N(x) - (I - P(z)))} (1 - y(n)) \right) \\ &\geq \left(\prod_{n \in (P(x) - P(z))} (1 - y(n)) \right) \left(\prod_{n \in (N(x) - (I - P(z)))} y(n) \right) \end{aligned}$$

Ainsi $\forall z \in \{0, 1\}^I$ on établit

$$\mathbb{P}(\{(x, y) \in \Omega / \mathbb{P}(x, y) \geq \mathbb{P}(z, y)\}) = PLemme_{\|z-x\|_1}$$

($PLemme_r$ correspond au lemme pour $p(s) = sC(s)$) Par rapport à l'algorithme glouton, ici, un voxel avec une forte concordance pourra être classé dans $N(x)$ au profit d'un voxel avec une concordance moins forte mais qui permet une meilleure probabilité globale.

Pour fixer les idées, sur l'exemple-jouet où : la grille de voxel est une grille 2D de 5 par 5 (avec 5 voxel dans $P(x)$ et 20 $N(x)$) et si $\forall u C(u) = 2$, on obtient :

$$\begin{aligned}
- ps_0 &= \left(\int_{0.5}^1 2udu \right)^{25} \approx 0.0007525434582 \\
- ps_1 &\approx \left(\int_{0.5}^1 2u \left(\int_{1-u}^1 2vdv \right)^8 du \right)^5 \approx 0.4814695055 \\
- PLemme_3 &\approx 0.9
\end{aligned}$$

Un des intérêts de la présente modélisation est de permettre d'étudier les propriétés des algorithmes de la famille étudiée, même ceux qu'on ne peut pas exécuter du fait de leur complexité.

7 Conclusion et perspectives

Dans cet article, nous avons proposé une modélisation mathématique probabiliste du problème de la reconstruction 3D qui permet d'étudier a priori les propriétés d'une famille d'algorithmes de reconstruction 3D. Nous avons montré que cette approche se justifie par le fait que les données utilisées par ces algorithmes vérifient certains invariants statistiques. Ensuite, nous avons montré pour quelques algorithmes qu'il y a bien compatibilité entre les propriétés prévues par la modélisation et les propriétés observés lors d'expériences effectuées sur des jeux de données réelles. Cela justifie le fait d'étudier théoriquement des reconstructions qui ne sont pas calculables dans la pratique, comme la reconstruction de la surface la plus probable.

Dans de futurs travaux, nous chercherons à améliorer notre modélisation statistique pour que tous ses paramètres soient calculables à partir des données. Cela permettrait de donner des garanties statistiques aux résultats des algorithmes de reconstruction de la famille étudiée dans cette article. Cela serait beaucoup moins fort que d'arriver à donner à des reconstructions 3D des garanties déterministes basées sur une modélisation physique et mathématique complète du processus de photographie. Néanmoins donner des garanties statistiques semble nettement plus accessible.

Références

- [1] Hiroshi Kawasaki, Ryo Furukawa, Ryusuke Sagawa, Yuya Ohta, Kazuhiro Sakashita, Ryota Zushi, Yasushi Yagi, Naoki Asada, Linear solution for oneshot active 3D reconstruction using two projectors, 3DPVT, 2010
- [2] Vladimir Kolmogorov, Ramin Zabih, Computing visual correspondence with occlusions using graph cuts, ICCV, 2001
- [3] G. Vogiatzis, P. Torr, R. Cippola, Multi-view Stereo via Volumetric Graph-Cuts, Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2005
- [4] S. Tran, L. Davis, 3D Surface Reconstruction Using Graph Cuts with Surface Constraints, Proceedings of the European Conference on Computer Vision, 2006
- [5] Y. Boykov, O. Veksler, R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001
- [6] V. Kolmogorov, Ramin Zabih, What Energy Functions Can Be Minimized via Graph Cuts ?, Proceedings of the European Conference on Computer Vision, 2002
- [7] C. Esteban, F. Schmitt, Silhouette and Stereo Fusion for 3D Object Modeling, Computer Vision and Image Understanding, 2004
- [8] M. Habbecke, L. Kobbelt, A Surface-Growing Approach to Multi-View Stereo Reconstruction, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007
- [9] Y. Furukawa, J. Ponce, Accurate, Dense, and Robust Multi-View Stereopsis, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007
- [10] Steven M. Seitz, Charles R. Dyer, Photorealistic Scene Reconstruction by Voxel Coloring, CVPR, 1997
- [11] K. N. Kutulakos and S. M. Seitz, A Theory of Shape by Space Carving, International Journal of Computer Vision, 2000
- [12] R. Guerchouche, O. Bernier, T. Zaharia, Reconstruction Volumétrique Multirésolution d'Objets 3D, RFIA, 2008
- [13] Vu Hoang Hiep, Renaud Keriven, Patrick Labatut, Jean-Philippe Pons, Towards high-resolution large-scale multi-view stereo, CVPR 2009
- [14] N. AMENTA, M. BERN, M. KAMVYSSELIS, A new voronoi-based surface reconstruction algorithm, SIGGRAPH, 1998
- [15] F. Chazal, Geometric inference for probability measures : extracting robust geometric information from noisy data, Journées STAR, 2010
- [16] Andrew Gelman, John B. Carlin, Hal S. Stern, Donald B. Rubin, Bayesian Data Analysis, 1993
- [17] R. Bhotika, D. J. Fleet, K. N. Kutulakos, A Probabilistic Theory of Occupancy and Emptiness, Proceedings of the European Conference on Computer Vision, 2002
- [18] S. Liu, D. B. Cooper, A Complete Statistical Inverse Ray Tracing Approach to Multi-View Stereo, CVPR, 2011
- [19] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, U. Thoennessen, On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery, CVPR, 2008