

Localisation et évaluation du diamètre et de la profondeur d'une cavité opératoire statique

I. Cheddadi¹, A. Getmanenko², I. Mostefaoui³, G. B. Nguyen⁴, M. Safa⁵,
D. Vicente⁶.

¹ LJLL, Université Paris 6

² IHÉS

³ Université de La Rochelle

⁴ LMJL, Université de Nantes

⁵ LMBA, Université de Bretagne Sud

⁶ MAPMO, Université d'Orléans

8ème Semaine d'Etude Maths-Entreprises.
SEME Orléans 20-24 Janvier 2014.

Sujet Proposé par l'entreprise MAQUET.
Correspondants : Cécile Louchet, Aurélien Gauvin.

Remerciements

Nous tenons à remercier l'entreprise Maquet et son représentant Aurélien Gauvin pour nous avoir proposé ce projet et pour le temps qu'il nous a consacré ; le laboratoire MAPMO qui a hébergé la SEME et en particulier Cécile Louchet qui a encadré le projet ; et enfin l'agence AMIES et le GdR Maths-entreprises qui sont à l'origine de cette opération et l'ont soutenue financièrement.

Table des matières

1	Introduction	4
1.1	Motivation et contexte	4
1.2	Objectif du rapport	5
1.3	Plan du rapport	5
2	Les données	5
2.1	La provenance des données	5
2.2	La structure des données	6
3	Cas d'une cavité à bord horizontal, sans ombre	6
4	Cas d'une cavité à bord incliné, sans ombre	8
5	Mise en oeuvre de l'algorithme d'identification des cavités	9
5.1	Algorithme de construction de la composante connexe d'un point vérifiant un critère donné	9
5.2	Identification de la cavité	10
5.3	Premiers résultats	10
5.4	Amélioration des résultats	10
6	Traitement des ombres	11
7	Conclusion et perspectives	14

1 Introduction

1.1 Motivation et contexte

Durant une opération chirurgicale, le médecin opère au sein de différentes couches du corps d'un patient, chacune ayant sa propre épaisseur et sa propre réflectivité. Un dispositif efficace d'éclairage de la table d'opération doit envoyer la lumière exactement sur la plaie chirurgicale et, dans la mesure du possible, éviter d'éclairer la peau autour d'elle afin de ne pas éblouir le chirurgien par l'excès de lumière réfléchi.

La société Maquet cherche à développer une technologie capable de contrôler de manière autonome la position et la forme du faisceau de lumière projeté sur la plaie chirurgicale.

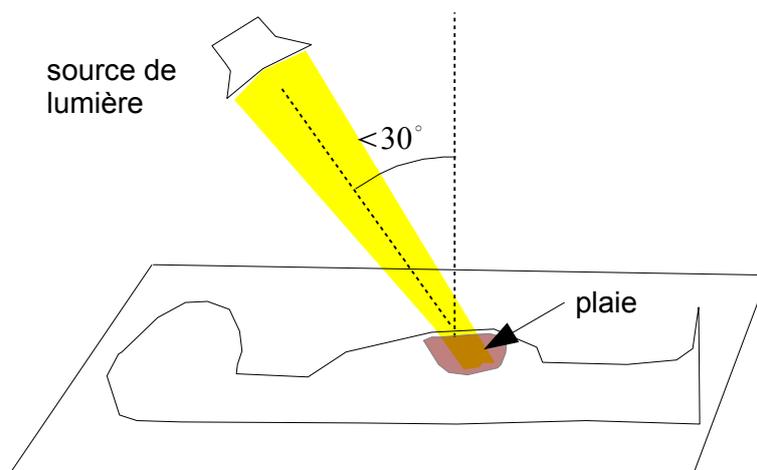


FIGURE 1 – La table opératoire, la plaie, et la source de la lumière ; dans ce cas précis, le faisceau de lumière est trop étroit et n'éclaire pas la totalité de la plaie.

Le problème technologique et algorithmique à résoudre est donc de localiser la plaie par rapport à la source de lumière. En conditions normales d'utilisation, cette source se trouve approximativement à la distance de 1 mètre de la table opératoire, et l'angle du faisceau par rapport à la verticale est inférieur à 30° . Au cours de l'opération la source de la lumière ainsi que le patient peuvent être fréquemment déplacés, ce qui nécessite une nouvelle détermination de leurs positions relatives une fois par seconde.

C'est pourquoi la société Maquet a proposé d'utiliser l'information donnée par un matériel de type Kinect. Un appareil Kinect contient une source de lumière infrarouge et une caméra digitale dans la même fréquence, capable de mesurer la distance de l'appareil à l'obstacle le plus proche. La scène 3D est représentée sous la forme d'une matrice 2D de profondeurs, la valeur de chaque pixel correspondant à la distance entre un objet et la caméra infrarouge. Cette technologie est actuellement utilisée dans la console de jeux vidéo Xbox 360 pour enregistrer les mouvements des joueurs par rapport à la console. Maquet propose d'installer un appareil Kinect sur la source de lumière chirurgicale.

1.2 Objectif du rapport

Dans ce rapport nous proposons une solution algorithmique élémentaire pour le problème de localisation de la plaie chirurgicale à partir de l'image Kinect.

1.3 Plan du rapport

Dans la Sec. 2, nous détaillons les formats informatiques des images de la plaie dans les champs visible et infrarouge.

Ensuite, nous discutons de l'algorithme dans le cas où il n'y a pas d'ombre projetée sur la table d'opération. La Sec. 3 s'intéresse au cas particulier où la surface autour de la cavité est horizontale. Dans la Sec. 4 nous traitons le cas plus général où cette surface est inclinée.

La Sec. 5 détaille la mise en oeuvre des algorithmes et l'évaluation de leur efficacité.

La Sec. 6 explique une méthode de suppression des ombres afin d'employer l'algorithme sur des images réalistes.

2 Les données

2.1 La provenance des données

L'appareil qu'on envisage de connecter à un ordinateur pour gérer l'illumination de la plaie chirurgicale comporte une lampe et une caméra digitale opérant en lumière visible, et le dispositif Kinect qui contient un émetteur et une caméra infrarouges. La distance entre la source de lumière et les caméras est de l'ordre de 2 cm, la distance entre les caméras et la plaie chirurgicale est de l'ordre d'1 m. A cause de cette distance entre la source de lumière infrarouge et les caméras, il résulte des ombres dans les images enregistrés (voir Fig..2, gauche).

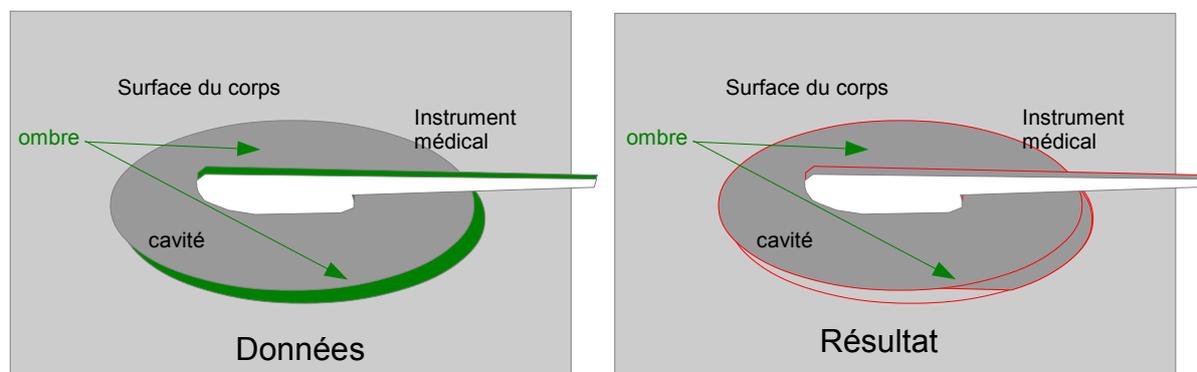


FIGURE 2 – Ombre sur l'image Kinect (gauche) et le résultat du retrait de l'ombre par l'algorithme expliqué à la Sec. 6, avec un artefact possible (droit).

Les deux types d'image sont transmis sous forme de fichiers au format JPEG en RGB sur 256 niveaux. Dans le cas de l'image Kinect, la distance entre la caméra infrarouge et l'objet (corps du patient) est représentée en niveau de gris (tous les canaux RGB ont même valeur), les pixels noirs représentant les points les plus éloignés. Les points appartenant à l'ombre, auxquels la Kinect n'arrive pas à attribuer une distance, sont représentés en vert, ce qui nous permet de les détecter facilement (les canaux RGB n'ont pas tous la même valeur) pour enlever l'effet de l'ombre de l'image, voir Sec. 6.

L'entreprise Maquet nous a indiqué que le niveau de bruit sur les mesures de profondeur est estimé inférieur à 10%.

L'entreprise Maquet nous a fourni des images simulées de table d'opération (voir Sec. 5.3), dans lesquelles l'épaisseur de l'ombre est de l'ordre de 5 à 10 % de la taille de l'image et jusqu'à 30 % de la taille de la cavité.

La Fig. 2, gauche montre une image schématique des données fournies.

On fera l'hypothèse que la cavité est plus profonde que la surface du corps autour. Donc nous n'avons pas traité le cas où les organes sortent de la cavité.

Par contre, nous devons traiter le cas où des instrument médicaux ainsi que les mains d'un médecin sont situés au-dessus de la cavité opératoire et où la forme de la cavité doit être détectée en dépit de la présence des ces éléments sur l'image Kinect. C'est l'une des difficultés principales que nous avons abordé.

Dans ce travail nous avons jugé suffisant de n'analyser que l'image Kinect et de ne pas utiliser l'image dans le champs visible.

2.2 La structure des données

L'image infrarouge obtenue par le Kinect donne la matrice de profondeur $z(x, y)$, Fig.3, où (x, y) désigne les coordonnées d'un pixel, et où $z(x, y)$ prend ses valeurs dans $\mathbb{Z} \cup \{\text{indéfini}\}$; $z(x, y) = \text{indéfini}$ signifie que le point (x, y) appartient à l'ombre. Pour les images fournies par

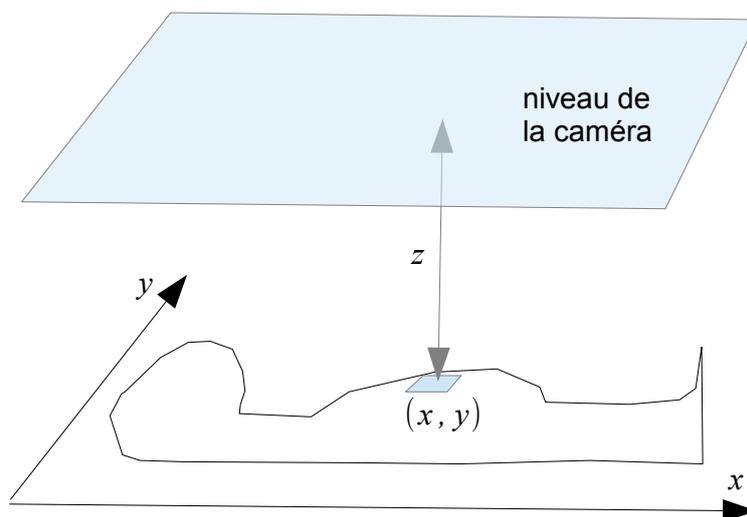


FIGURE 3 – Les données de la matrice de profondeur $z(x, y)$

Maquet, les coordonnées x, y sont les entiers $0 \leq x, y \leq 600$, et les valeurs de $z_{x,y}$ sont entre 0 et 255.

3 Cas d'une cavité à bord horizontal, sans ombre

Dans ce paragraphe, nous nous plaçons dans la situation où la surface entourant la cavité est orthogonale à l'axe de la caméra, et donc apparaît comme un région approximativement plate à distance constante de la Kinect. Mathématiquement, quitte à passer dans un système de coordonnées approprié, on peut donc supposer pour le reste de Sec. 3 que la surface est horizontale et que l'axe de la caméra est vertical.

Par ailleurs, nous ferons l'hypothèse que le centre de l'image appartient à la cavité. Il peut arriver cependant que le point situé exactement au centre de l'image soit plus élevé que la surface

autour de la plaie, soit à cause du bruit, soit parce que ce point correspond à un instrument médical. Pour cet raison, on réitère l’algorithme décrit ci-dessous plusieurs fois à partir d’un point initial choisi comme le point central de l’image décalé d’un vecteur aléatoire $(\Delta x, \Delta y)$ avec $-30 \leq \Delta x \leq 30$, $-30 \leq \Delta y \leq 30$. On désigne par (x_*, y_*) le point initial situé au milieu de l’image.

L’idée intuitive de l’algorithme est de remplir la cavité d’eau jusqu’au niveau où l’eau commence à déborder. Le niveau de l’eau juste avant de déborder sera notre représentation de la forme de la cavité.

Nous aurons besoin de la définition suivante : Si A est un sous-ensemble de \mathbb{Z}^2 , on l’appelle *connexe* si pour tous les couples de points $(x_d, y_d), (x_f, y_f) \in A$ il existe un chemin dans A qui les relie, c’est à dire un nombre n de points $(x_1, y_1), \dots, (x_n, y_n) \in A$ tels que $|x_j - x_{j-1}| + |y_j - y_{j-1}| = 1$, $1 \leq j \leq n + 1$, où on prends $(x_0, y_0) = (x_d, y_d)$ et $(x_{n+1}, y_{n+1}) = (x_f, y_f)$.

Plus précisément, pour un paramètre convenable H (“hauteur de l’eau”), $0 \leq H \leq 255$, on calcule la composante connexe contenant (x_*, y_*) de l’ensemble $\{(x, y) : z_{xy} < H\}$. On désignera cette composante connexe par $CC_H \subset \mathbb{Z}^2$ (la dépendance de (x_*, y_*) sera supprimée de la notation).

On appelle le *niveau critique* H_{cr} (“hauteur critique”) la valeur de H minimale pour laquelle CC_H contient un des points au bord de l’image, i.e. un point (x, y) avec $x = 0$ ou $y = 0$ ou $x = 600$ ou $y = 600$.

On prend donc $CC_{H_{cr}-1}$ comme une approximation de la forme de la cavité.

Étant donnée un valeur de H et un point initial (x_*, y_*) , on peut calculer l’ensemble CC_H au moyen d’un algorithme standard de programmation dynamique, voir fr.wikipedia.org/wiki/Programmation_dynamique. Pour chercher le niveau critique H_{cr} nous avons exploré par dichotomie le segment $[0, 255]$ des valeurs possibles de H .

Une proposition de raffinement. Cet approche marche bien pour les données simulées que nous avons reçues de l’entreprise. Dans une situation plus réaliste on peut imaginer une cavité entourée d’un région plus basse que la surface du corps ; une section verticale d’une telle situation est montrée sur la Fig. 4.

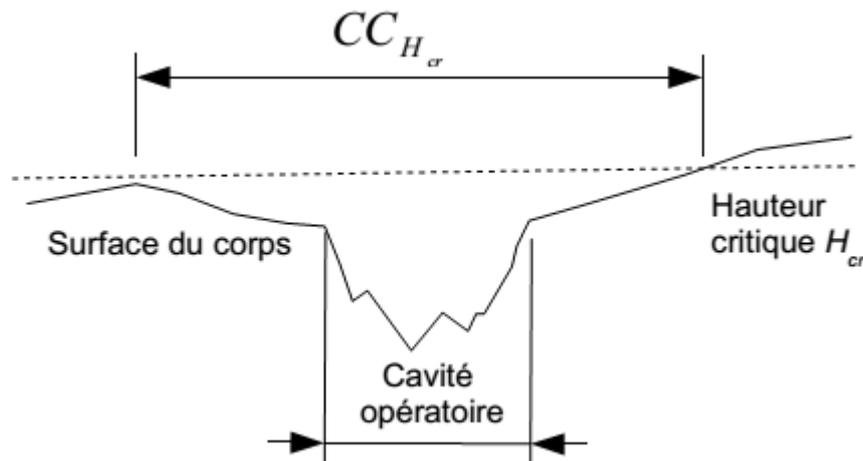


FIGURE 4 – Un cas où $CC_{H_{cr}}$ surestime la taille de la cavité chirurgicale.

Pour éviter ce genre d’effets, nous proposons de calculer et de noter une nombre suffisant de paires $(H_j, \#CC_{H_j})$, où $\#(\cdot)$ désigne le nombre des éléments d’un ensemble. Nous nous attendons à ce que le comportement des $(H_j, \#CC_{H_j})$ suive le comportement indiqué dans la figure 5. Un

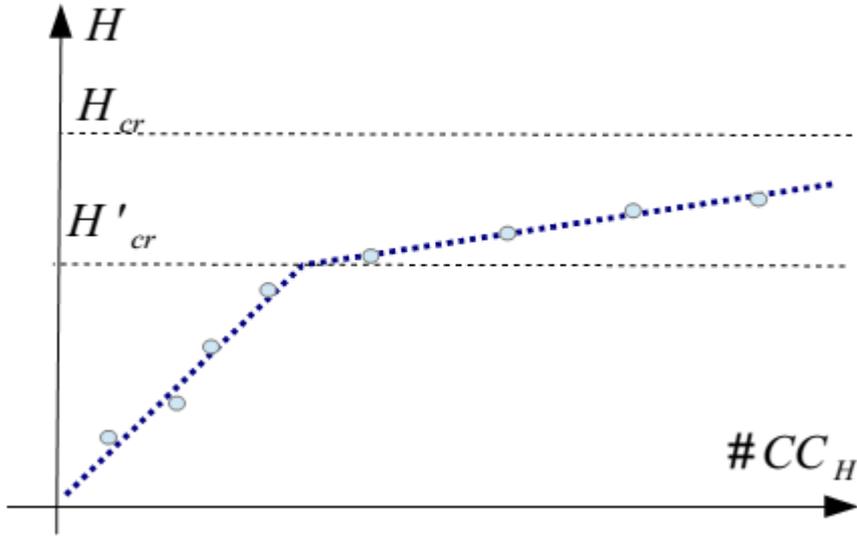


FIGURE 5 – Détection d’un bord plus bas que le reste du tissu : comportement attendu de la hauteur en fonction de la taille de la composante connexe correspondante.

tel comportement rendrait raisonnable l’approximation (régression statistique) par une fonction linéaire par morceaux :

$$\begin{cases} \#CC_H \sim k_1 H + b_1, & H < H'_{cr}, \\ \#CC_H \sim k_2 H + b_2, & H \geq H'_{cr}, \end{cases}$$

à cinq paramètres $k_1, b_1, k_2, b_2, H'_{cr}$ à déterminer, soumis à la contrainte $k_1 H'_{cr} + b_1 = k_2 H'_{cr} + b_2$. Nous attendons que $CC_{H'_{cr}-1}$ donne une approximation plus précise de la forme de la cavité que $CC_{H_{cr}}$.

En l’absence de données réalistes, nous n’avons pas testé cette approche.

4 Cas d’une cavité à bord incliné, sans ombre

Selon le représentant de l’entreprise, l’angle entre l’axe de la caméra et la surface du corps (hors de la cavité) peut dévier jusqu’à 30 degrés de l’angle droit. En ce cas notre méthode du niveau critique exposée dans Sec.3 sous-estimera la taille de la cavité (voir Fig. 6). L’idée immédiate est de faire tourner le corps (numériquement) de 20° en chacune des quatre directions gauche bas droite haut, estimer $CC_{H_{cr}}$ pour les quatre images transformées, et prendre la réunion des quatre ensembles (proprement transformés).

Un procédé à peu près équivalent mais beaucoup moins coûteux est d’effectuer une transformation affine $z(x, y) \mapsto z(x, y) + \varepsilon_1 \sin 20^\circ x + \varepsilon_2 \sin 20^\circ y$, où $(\varepsilon_1, \varepsilon_2)$ parcourt les quatre valeurs $(1, 0), (0, 1), (-1, 0), (0, -1)$. La réunion des quatre ensembles $CC_{H_{cr}}$ ainsi calculés nous donnera l’approximation de la cavité opératoire. Une inclinaison de 20° pour la transformation affine est une possibilité raisonnable.

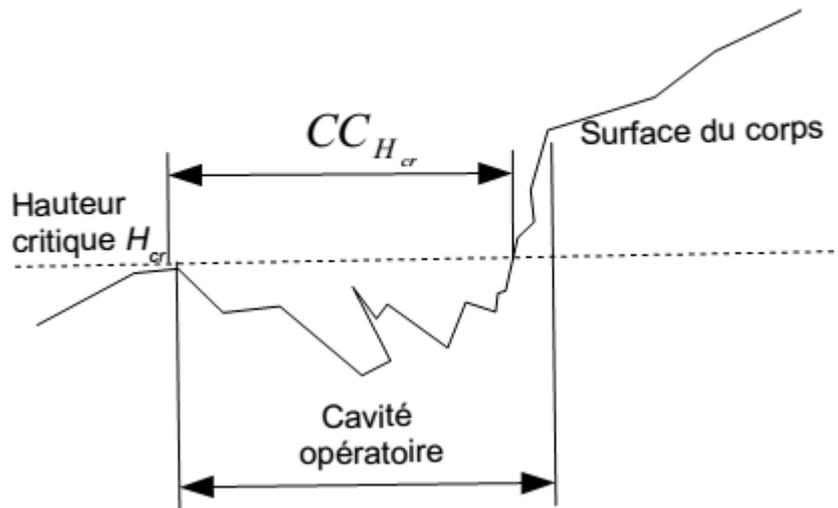


FIGURE 6 – Cas où la cavité est inclinée. L’algorithme risque de sous-estimer la taille de la cavité.

5 Mise en oeuvre de l’algorithme d’identification des cavités

5.1 Algorithme de construction de la composante connexe d’un point vérifiant un critère donné

Soit $f(p)$ une fonction booléenne qui à un pixel associe *vrai* si le critère est vérifié et *faux* sinon. L’algorithme suivant construit la composante connexe (CC) contenant un point p_0 vérifiant le critère fixé :

```
#initialisation
si f(p0) est faux
    la composante connexe est vide
    arrêt de l’algorithme
sinon,
    ajouter p0 à la CC
    marquer p0
    ajouter ses voisins à la liste des points à visiter
    les marquer
fin si
#boucle sur le voisinage
tant que la liste des points à visiter n’est pas vide :
    choisir un point p
    le retirer de la liste
    si f(p)
        ajouter p à la CC
        ajouter ses voisins non marqués à la liste des points à visiter
        les marquer
    sinon
        continuer
```

```
fin si
fin tant que
```

Le coût de cet algorithme est linéaire par rapport au nombre de pixels.

5.2 Identification de la cavité

Nous supposons qu'un point à l'intérieur de la cavité est connu ; pour fixer les idées, le milieu de l'image (ce qui reste cohérent par rapport à une utilisation en conditions réelles, dans la mesure où le chirurgien place la lampe de manière à ce qu'elle éclaire la cavité). Nous expliquons plus loin comment lever cette limitation.

Comme expliqué précédemment, la méthode consiste à identifier le niveau h_{max} à partir duquel un liquide déborderait de la cavité. Soit $z(p)$ la profondeur correspondant à un pixel (donnée fournie directement par la kinect) et h une hauteur fixée ; nous utilisons le critère suivant pour le calcul de la composante connexe :

$$f_h(p) = \begin{cases} vrai & \text{si } z(p) < h \\ faux & \text{sinon} \end{cases}$$

Dans notre contexte, nous considérons que la cavité “déborde” à une hauteur donnée h lorsque la composante connexe CC_h obtenue avec le critère $f_h(p)$ touche le bord. La hauteur est un entier de l'intervalle $[0, 255]$ (avec les données fournies) et la hauteur limite h_{max} vérifie $CC_{h_{max}+1}$ touche le bord, et $CC_{h_{max}}$ ne touche pas le bord. Une illustration de cette méthode est donnée dans la section suivante. La hauteur h_{max} est déterminée par une méthode de dichotomie, en partant de l'intervalle $[z(p_0), 255]$; il est tout à fait possible que h_{max} ne soit pas définie, dans le cas où la composante connexe $CC_{z(p_0)}$ touche déjà le bord. Dans ce cas, il faut déterminer un autre point initial. Nous avons choisi de parcourir aléatoirement un voisinage du point initial ; pour cela nous avons utilisé la fonction `rand()` de la librairie standard C++ (<http://www.cplusplus.com/reference/cstdlib/rand/>).

5.3 Premiers résultats

Pour des raisons légales, il n'est pas possible pour l'instant de réaliser des cartes de profondeur à partir de la kinect sur de vraies tables d'opérations, aussi, Maquet nous a-t-il fourni une batterie de cas-tests (Fig. 7) où varient un certain nombre de paramètres :

- l'orientation du plan dans lequel se situe la cavité par rapport à la caméra
- la présence d'une serviette rouge à l'intérieur de la cavité (pour imiter des organes qui dépasseraient de la cavité)
- la présence d'instruments sur le bord de la cavité
- l'aspect plus ou moins désordonné du pourtour de la cavité

Commençons par illustrer la méthode sur un premier cas test (Fig. 8) : la méthode de dichotomie donne $h_{max} = 237$ et effectivement, la composante connexe CC_{238} touche le bord mais pas la composante connexe CC_{237} .

Cette méthode appliquées à tous les autres cas-tests donne des résultats satisfaisants (Fig. 9) : la taille de la composante connexe est toujours de l'ordre de la taille de la cavité, ce qui est une information suffisante pour notre application.

5.4 Amélioration des résultats

Pour améliorer ces résultats, nous appliquons la méthode exposée en Sec.4, qui consiste à appliquer une transformation affine à la carte de profondeur de manière à “rectifier” son

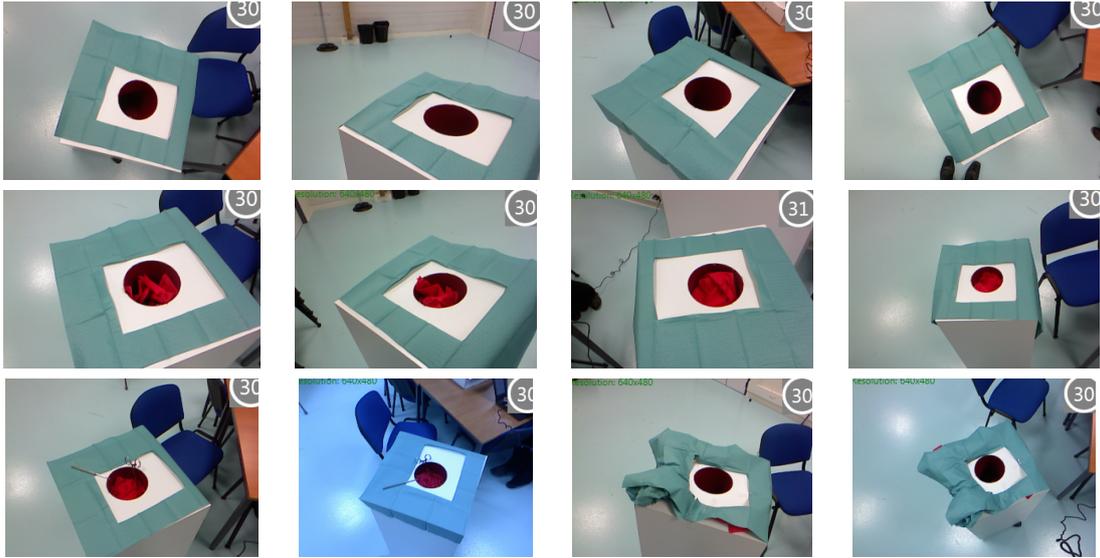


FIGURE 7 – Cas-tests fournis par Maquet. Images dans le champ visible.

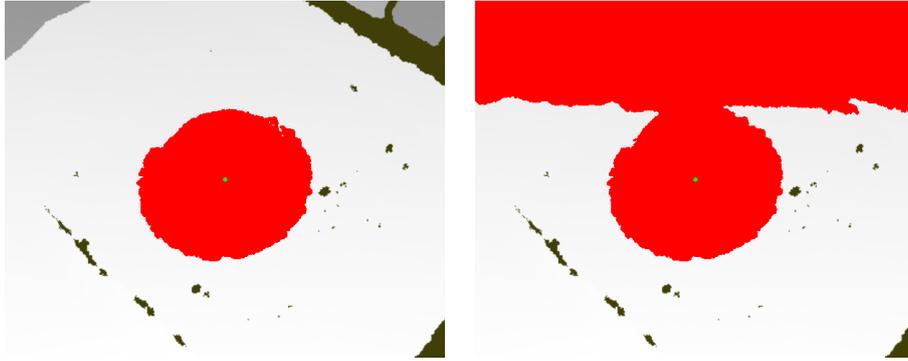


FIGURE 8 – Illustration de la méthode : à gauche la composante connexe CC_{237} , à droite la composante connexe CC_{238} qui touche le bord. On identifie ainsi $h_{max} = 237$ et la cavité est identifiée comme la composante connexe $CC_{h_{max}}$.

orientation par rapport à l'axe de la caméra, ce qui est illustré dans la Fig. 10. Ici, l'axe de rotation et l'angle sont déterminés à la main, on peut envisager une procédure automatique pour les fixer de manière optimale. On peut également envisager simplement d'effectuer quatre rotations suivant les deux axes x et y , de $\pm 20^\circ$.

6 Traitement des ombres

Le retrait des artefacts de l'image est un problème abondamment étudié, voir par exemple <http://en.wikipedia.org/wiki/Inpainting>. Dans le cas présent, le coût de calcul doit être suffisamment faible pour que l'algorithme se termine en moins d'une seconde, ce qui nous a amenés à essayer les algorithmes les plus simples possibles au détriment de la fidélité aux détails de l'image initiale.

Nous supposons que la kinect renvoie une valeur en niveaux de gris pour les zones où elle

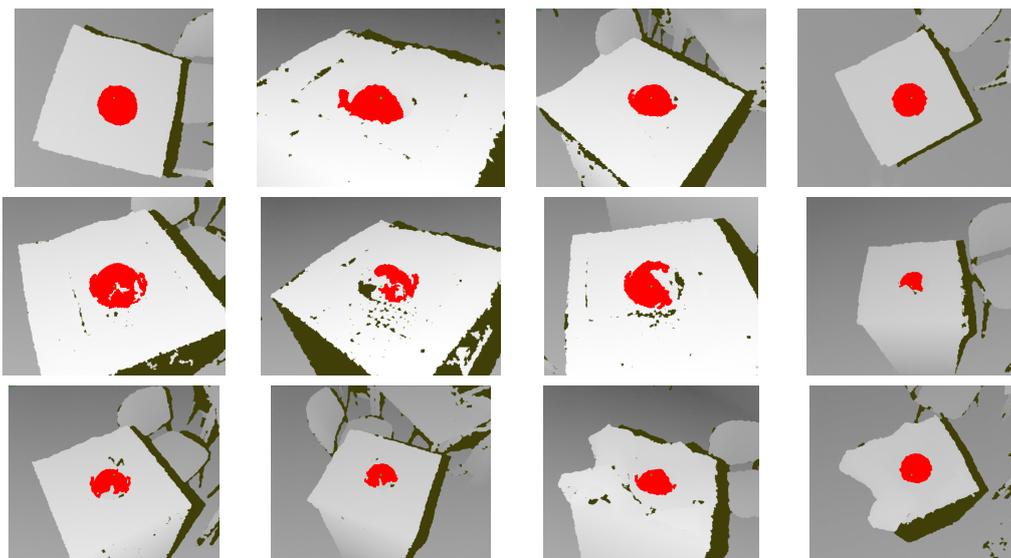


FIGURE 9 – Identification de la cavité : en rouge, la composante connexe CCh_{max} (voir texte), superposée à la carte de profondeur donnée par la kinect.

arrive à mesurer la profondeur, et trois valeurs différentes en RGB (en pratique, cette couleur est brun vert foncé), ce qui permet à l’algorithme de détecter automatiquement les zones d’ombre.

Une approche possible serait de remplacer une valeur $z(x, y) = \text{indéfini}$ par la valeur $z(x', y')$, où (x', y') est le pixel le plus proche de (x, y) parmi les pixels hors de l’ombre, c’est-à-dire, ayant $z(x', y') < \text{indéfini}$. Cependant cette approche mettrait une partie importante de l’ombre (proche de la région de l’image moins profonde) hors de la cavité, alors qu’on s’attend à ce que la région de l’ombre appartienne en réalité à la cavité.

L’algorithme que nous avons mis en place consiste à parcourir successivement les lignes de l’image, et à remplacer les zones d’ombre par le minimum des valeurs de profondeur les plus proches (Fig. 11). C’est-à-dire, si $z_{l+1,y} = \dots = z_{r-1,y} = +\infty$ mais $z_{l,y}, z_{r,y} < +\infty$, on redéfinit $z_{l+1,y}, \dots, z_{r-1,y}$ comme $\min\{z_{l,y}, z_{r,y}\}$.

Cela peut conduire à des artefacts dans les cas où l’ombre est orientée horizontalement, dans la direction des lignes de l’image, Fig. 2 : il y aurait alors le risque que l’algorithme propage artificiellement une profondeur basse, ce qui aurait comme conséquence de perturber l’algorithme de recherche de la cavité (penser à un canal sortant de la cavité : il serait alors impossible de la remplir, voir Fig.2).

Une première possibilité est de parcourir l’image dans deux directions différentes (par exemple suivant les lignes et suivant les colonnes), à appliquer la méthode précédente, puis à prendre le maximum des valeurs obtenues pour remplacer l’ombre. Ainsi on échapperait au risque de “creuser” un canal dans la cavité, mais on perdrait en précision sur la position du bord de la cavité.

Une deuxième solution est basée sur l’idée suivante : si on est capable d’identifier la direction principale d’une zone d’ombre, en parcourant l’image dans la direction perpendiculaire, on évitera le problème évoqué plus haut. D’un point de vue pratique, on pourrait adapter notre algorithme de recherche de composante connexe pour identifier chaque zone d’ombre, puis d’en identifier la direction principale (par exemple en identifiant la meilleure ellipse au sens des moindres carrés).

Enfin, on pourrait faire appel à des méthodes plus sophistiquées qui exploitent non seulement

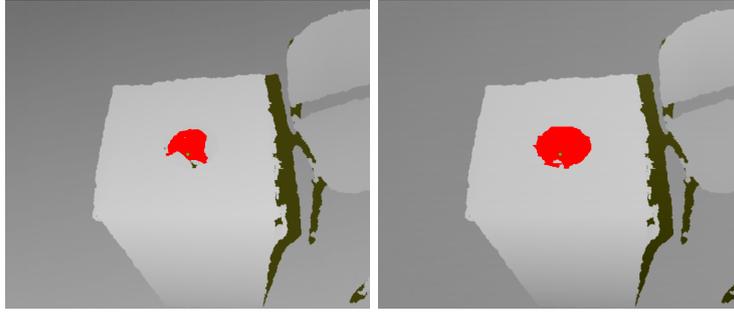


FIGURE 10 – Effet de l’application de la transformation affine $f(i, j) = 0.05(N/2 - i)$ sur une des images : à gauche, on a appliqué l’algorithme présenté à la Sec.3 sans la transformation ; à droite avec la transformation.

Ici, i est le numéro de ligne dans l’image, N est le nombre de pixels dans la direction verticale, donc $N/2 - i$ correspond à une ordonnée qui vaudrait zéro au milieu de l’image. Ainsi, cette transformation va faire “remonter” le haut de l’image, de manière à faire basculer le liquide vers les parties de la cavité qui n’étaient pas atteintes par la composante connexe.

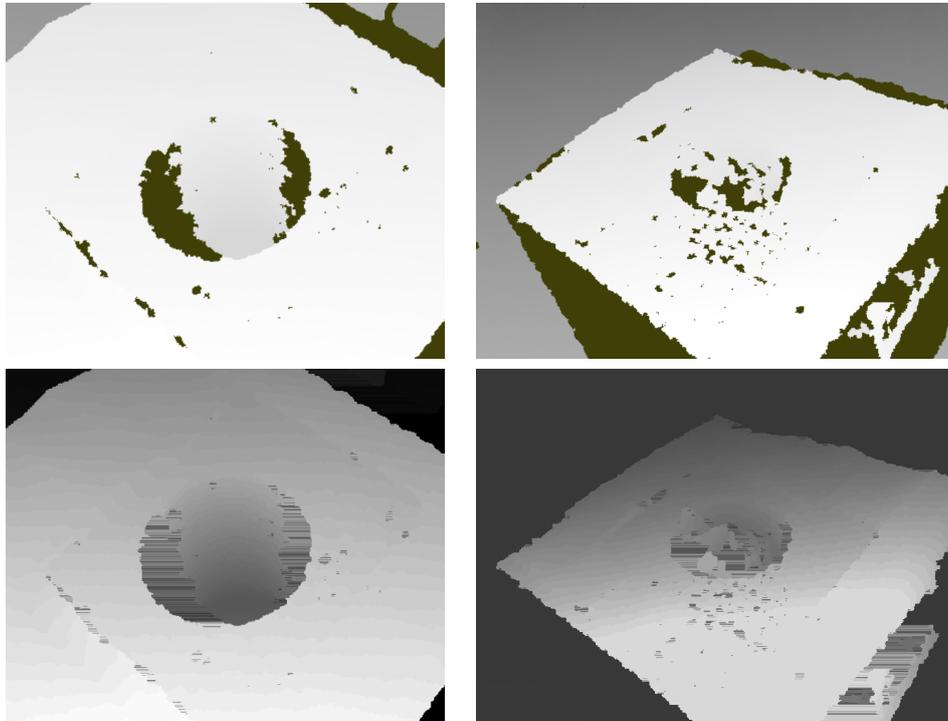


FIGURE 11 – Exemple d’application de l’algorithme de suppression des ombres à partir de cartes de profondeur venant de la kinect ; l’image est parcourue ligne par ligne (voir texte). Haut : images originales ; bas : images transformées.

une image statique, mais une séquence temporelle d’images ; ce type de méthode a été mis en place pour la kinect, par exemple dans le projet

<https://code.google.com/p/kinect-depth-map-inpainting-and-filtering/>

7 Conclusion et perspectives

L'objectif de ce travail, qui s'est déroulé dans le cadre de la 8ème Semaine d'études Maths-Entreprise, est de développer un outil permettant de détecter la cavité opératoire lors d'une opération chirurgicale. La localisation en espace de cette cavité permet de l'éclairer d'une manière optimale en modifiant l'intensité et la forme du faisceau lumineux projeté sur la plaie chirurgicale. En utilisant les images issues d'un Kinect et qui représentent la carte de profondeur de la plaie opératoire, nous avons développé un algorithme permettant de détecter le bord de la cavité opératoire. L'idée de cet algorithme, basé sur la programmation dynamique et que nous avons codé en C++, est de remplir la cavité d'eau jusqu'au niveau où l'eau commence à déborder. Le niveau de l'eau juste avant de déborder est notre représentation de la forme de la cavité. Nous avons appliqué cet algorithme sur des images fournies par l'entreprise Maquet et qui imitent les vraies tables d'opérations. Les résultats obtenus sont très satisfaisants dans le cas où dans les images issues de Kinect ne figurent pas des ombres. Afin de traiter les zones d'ombres dans les images de Kinect, nous avons mis en place un algorithme qui permet de traiter les zones d'ombres. Cet algorithme consiste à parcourir successivement les lignes de l'image et à remplacer les zones d'ombre par le minimum des valeurs de profondeur les plus proches.

L'algorithme que nous avons développé nécessite un point initial qui se trouve à l'intérieur de la cavité. Il serait intéressant de pouvoir initialiser le programme avec un point quelconque de l'image. Par ailleurs et afin de mieux gérer les ombres, il est possible d'améliorer l'algorithme présenté la section 6 en utilisant des méthodes basées sur le filtrage et le traitement d'image.