

# Holographie numérique de particules : amélioration de l'algorithme de dépouillement par optimisation

Jérôme Gire, Christophe Ducottet

► **To cite this version:**

Jérôme Gire, Christophe Ducottet. Holographie numérique de particules : amélioration de l'algorithme de dépouillement par optimisation. CMOI/FLUVISU 2009, Nov 2009, Reims, France. Session 3 : Holographie, 2009. <ujm-00511217>

**HAL Id: ujm-00511217**

**<https://hal-ujm.archives-ouvertes.fr/ujm-00511217>**

Submitted on 24 Aug 2010

**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.

# Holographie numérique de particules : amélioration de l'algorithme de dépouillement par optimisation

*Jérôme GIRE, Christophe DUCOTTET*

*Laboratoire Hubert Curien, CNRS UMR 5516, Université de Saint-Etienne, Université de Lyon  
18, rue du Pr Benoit Laurus, F-42000 Saint-Etienne, France*

## 1. Introduction

L'holographie en ligne est une technique d'imagerie 3D qui est utilisée depuis de nombreuses années notamment dans le domaine de la caractérisation des écoulements fluides [8, 11]. Elle permet notamment le positionnement 3D et la mesure de taille de petits objets à partir de l'acquisition d'une seule image 2D [4]. La version numérique de cette technique réalise un enregistrement direct sur un capteur et un dépouillement numérique des images-hologramme ne nécessitant pas de reconstruction optique [3, 5]. Le montage et la mise en œuvre particulièrement simple de la technique la rendent compatible avec un environnement industriel complexe.

Un grand nombre d'approches ont été proposés pour le dépouillement d'hologrammes de microparticules. La plupart sont basées sur une reconstruction numérique simulant la diffraction de l'hologramme [6, 7, 2]. Notre équipe a récemment proposé une approche par optimisation de type « problèmes inverses » [9]. Cette approche consiste à rechercher les paramètres de position et de taille de chaque particule en minimisant l'écart entre l'hologramme enregistré et le modèle de cet hologramme. Elle permet notamment d'augmenter la précision sur la localisation longitudinale de la particule et d'augmenter la taille du champ accessible au-delà de la taille du capteur [10]. Elle permet également de s'affranchir du problème des images fantôme et de relaxer les contraintes d'échantillonnage liées au capteur numérique [1]. En revanche, le temps de traitement des hologrammes est beaucoup plus important qu'avec une approche « classique » qui procède à une reconstruction du volume des particules en simulant la diffraction de l'hologramme.

Nous présentons dans cette communication une amélioration de l'algorithme de dépouillement par optimisation qui a pour objectif de réduire la complexité de cet algorithme. Nous évaluons ensuite les performances de cette nouvelle version de l'algorithme à partir d'expérimentations numériques menées sur des hologrammes synthétiques.

Le paragraphe 2 présente d'abord l'algorithme de dépouillement initial après avoir rappelé le modèle de formation d'un hologramme. Le paragraphe 3 présente les améliorations que nous proposons qui sont ensuite évaluées dans le paragraphe 4.

## 2. L'algorithme de dépouillement par approche « problème inverse »

Nous nous plaçons dans le contexte de l'holographie numérique de micro-objets dont la taille est comprise entre 10 et 500 $\mu\text{m}$ . Cette technique permet notamment le positionnement 3D et la mesure de taille de ces objets à partir de l'acquisition d'une seule image 2D (hologramme). La configuration expérimentale adaptée à cette situation est le montage en ligne de Gabor (Figure 1). Dans ce cas, l'onde de référence et l'onde éclairant les particules sont confondues et l'on peut considérer que l'intensité enregistrée sur l'hologramme correspond à la figure de diffraction des particules. Cette figure est enregistrée directement sur un capteur discret de type CCD ou CMOS. L'image obtenue est dite hologramme numérique.

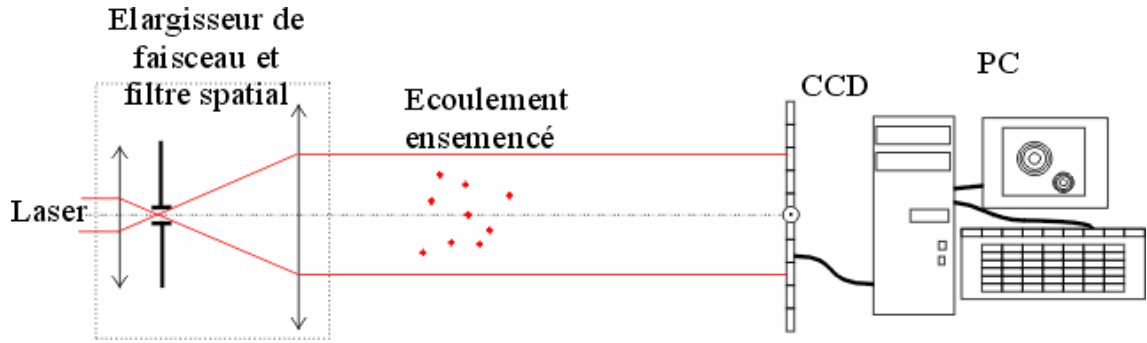


Figure 1: Montage en ligne de Gabor

Le dépouillement de cet hologramme consiste à détecter toutes les particules du volume d'étude et à déterminer pour chacune d'elle sa position 3D et sa taille. Nous présentons dans ce paragraphe l'algorithme de dépouillement par optimisation basé sur une approche « problèmes inverses ». Connaissant le modèle d'enregistrement de l'hologramme, l'approche problème inverse revient à déterminer les paramètres de toutes les particules du volume, de manière à minimiser l'écart quadratique entre l'hologramme numérique et le modèle. Nous présentons le modèle d'enregistrement, la formulation en problème inverse et l'algorithme associé.

### Modèle d'enregistrement de l'hologramme

Dans le contexte de la diffraction de Fresnel, l'intensité enregistrée sur l'hologramme peut être approximée comme la somme des contributions des intensités de chaque particule et d'un bruit de fond [9]. Un ensemble de  $n$  particules sphériques est décrit par les paramètres  $\{x_k, y_k, z_k, r_k\}_{k=1, \dots, n}$  représentant respectivement les coordonnées et le rayon de chaque particule. Le modèle d'enregistrement de l'hologramme est donné par :

$$m[i, j] = I_0 - \sum_{k=1}^n \alpha_k g_k[i, j] + I_{bg}[i, j]$$

où  $I_0$  désigne la contribution de l'onde plane incidente,  $g_k[i, j]$  désigne le modèle de la figure de diffraction de la particule,  $\alpha_k$  le facteur de pondération de ce modèle et  $I_{bg}[i, j]$  le bruit de fond. L'expression du modèle de diffraction d'une particule est donné par :

$$g_k[i, j] = \frac{\pi r_k^2}{\lambda z_k} J_{1c} \left( \frac{2\pi r_k \sqrt{(i\Delta\xi - x_k)^2 + (j\Delta\xi - y_k)^2}}{\lambda z_k} \right) \sin \left( \frac{\pi((i\Delta\xi - x_k)^2 + (j\Delta\xi - y_k)^2)}{\lambda z_k} \right)$$

où  $\lambda$  désigne la longueur d'onde,  $\Delta\xi$  le pas d'échantillonnage du capteur et  $J_{1c}$  la fonction de Bessel cardinal d'ordre 1.

### Formulation en problème inverse

L'hologramme numérique obtenu après acquisition est une matrice de données notée  $d[i, j]$ . Le problème inverse consiste à trouver la valeur  $I_0^+$ , les valeurs  $\{\alpha_k^+\}_{k=1, \dots, n}$  et le jeu de paramètre des particules  $\{x_k^+, y_k^+, z_k^+, r_k^+\}_{k=1, \dots, n}$  qui minimisent la fonction de coût  $\mathcal{P}$  donnée par :

$$\mathcal{P} = \sum_w (m[i, j] - d[i, j])^2$$

où l'opérateur  $\sum_w$  représente une sommation pondérée sur les indices  $(i, j)$ . La fonction de pondération  $w[i, j]$  correspond à la dimension limitée du capteur. Elle vaut 1 sur chaque pixel mesuré et 0 ailleurs.

## Algorithme de dépouillement

L'algorithme proposé dans [9] procède de manière itérative en recherchant, à chaque itération, les paramètres de la particule la plus vraisemblable, puis en retirant la contribution de cette particule aux données initiales (gommage). Les paramètres de la particule sont estimés en deux étapes. La première étape de détection globale détermine une valeur initiale en explorant l'espace des paramètres. La deuxième étape d'optimisation locale ajuste les paramètres.

Ainsi, à une itération donnée  $k$ , les paramètres  $\{x_k^+, y_k^+, z_k^+, r_k^+\}$  de la  $k^{\text{ème}}$  particule sont obtenus en minimisant la pénalité  $\mathcal{P}_k$  donnée par :

$$\mathcal{P}_k = \sum_w (m_k[i, j] - d_k[i, j])^2$$

où  $m_k[i, j]$  est la contribution de la  $k^{\text{ème}}$  particule et  $d_k[i, j]$  est l'hogramme après soustraction des contributions des particules détectées avant l'itération  $k$  (hogramme résiduel). On a :

$$m_k[i, j] = I_k - \alpha_k g_k[i, j]$$

L'étape de détection globale est effectuée en maximisant le critère  $\mathcal{Q}_k$  défini par :

$$\mathcal{Q}_k = \frac{\left( \sum_w d_k[i, j] \tilde{g}_k[i, j] \right)^2}{\sum_w \tilde{g}_k^2[i, j]}$$

où  $\tilde{g}_k$  est la version de  $g_k$  centrée sur le capteur.

La maximisation est effectuée par recherche exhaustive dans l'espace des paramètres  $(x, y, z, r)$ . Une fois les valeurs initiales  $\{x_k^0, y_k^0, z_k^0, r_k^0\}$  déterminées, on peut calculer les valeurs initiales  $\{I_k^0, \alpha_k^0\}$  par optimisation linéaire de la pénalité  $\mathcal{P}_k$  pour  $(x, y, z, r)$  fixés.

A partir du jeu de paramètres initiaux  $\{x_k^0, y_k^0, z_k^0, I_k^0, \alpha_k^0\}$ , on calcule un ajustement précis des paramètres par optimisation locale non linéaire de la pénalité  $\mathcal{P}_k$ .

L'algorithme correspondant est donné ci-dessous.

### Algorithme 1

<b>Entrée :</b> $d[i, j]$	{ hogramme numérique }
<b>Sortie :</b> $\{x_k^+, y_k^+, z_k^+, r_k^+\}_{k=1, \dots, n}$	{ paramètre des particules }
$k \leftarrow 0$	
$d_0 \leftarrow d$	
$fin \leftarrow faux$	
<b>Tant que</b> (non fin)	
$(x_k^0, y_k^0, z_k^0, r_k^0) = \text{DéetectGlobale}(d_k)$	{ détection globale }
$(I_k^0, \alpha_k^0) = \text{OptimLinéaire}(d_k, k)$	{ valeurs initiales pour $I_k$ et $\alpha_k$ }
$(x_k^+, y_k^+, z_k^+, r_k^+, I_k^+, \alpha_k^+) = \text{OptimLocale}(d_k, k)$	{ optimisation locale particule $k$ }
<b>Si</b> $(\alpha_k > \epsilon)$ <b>alors</b>	{ critère d'arrêt }
$d_{k+1} \leftarrow d_k - m_k$	{ gommage }
$k \leftarrow k + 1$	
<b>Sinon</b>	
$fin \leftarrow vrai$	

**Fait**

### Complexité

Les deux parties les plus coûteuses en temps de calcul, sont les deux fonctions *DéetectGlobale* et *OptimLocale* qui sont répétées pour chaque particule détectées. La partie *OptimLinéaire* sera considérée dans la suite incluse dans la partie *OptimLocale*.

Pour la fonction *DétectionGlobale*, on définit un échantillonnage  $(z_p, r_p)_{p=1, \dots, n_{zr}}$  des paramètres  $(z, r)$ . Cet échantillonnage en  $n_{zr}$  couples  $(z, r)$  est choisi en fonction de la taille du volume d'étude et de la granulométrie des particules. Etant donné un couple de paramètres  $(z_p, r_p)$ , le critère  $Q_k$  peut être calculé à partir de transformées de Fourier rapides (FFT) sur une grille discrète de l'espace  $(x, y)$ . Ce calcul peut être considéré comme la corrélation normalisée entre l'hogramme numérique résiduel et un modèle de particule de paramètres  $(z_p, r_p)$ . Le pas d'échantillonnage est le même que celui du capteur, mais la taille du champ peut être plus grande que le capteur [10]. L'étape de détection globale consiste alors à rechercher dans cet espace de paramètres échantillonnés, le jeu de paramètres  $\{x_k^0, y_k^0, z_k^0, r_k^0\}$  qui maximise le critère  $Q_k$ . Si  $N$  désigne la taille du capteur, la complexité de cette fonction est donc donnée par :

$$C_1 = O(n_{zr} N^2 \log_2(N))$$

Pour la fonction *OptimLocale*, la complexité dépend de la taille du capteur et du nombre d'itérations  $n_{opt}$  de l'optimisation. Elle est donnée par :

$$C_2 = O(n_{opt} N^2)$$

Si  $n_p$  désigne le nombre de particules de l'hogramme, la complexité totale de l'algorithme est donc donnée par :

$$C = O(n_p n_{zr} N^2 \log_2(N)) + O(n_p n_{opt} N^2)$$

En pratique, avec notre implémentation, on constate que le temps d'exécution est d'environ 3 minutes par particules pour un hogramme de  $N=1024$  avec  $n_{zr} = 100$ .

### 3. Améliorations de l'algorithme

Dans l'algorithme initial, à chaque itération, la première étape de détection globale doit être effectuée pour déterminer les paramètres de la particule la plus vraisemblable de l'hogramme résiduel. Dans un deuxième temps, les paramètres de cette particule sont ajustés par optimisation. Plusieurs inconvénients résultent de cette procédure :

- La détection globale doit être faite à chaque itération alors que le résultat du calcul du critère  $Q_k$  pourrait être exploité pour identifier d'autres particules. Le temps de calcul est donc pénalisé.
- Lorsque des particules sont proches, le groupe qu'elles forment peut produire des fausses détections ou des erreurs d'estimation qui peuvent être difficile à diagnostiquer. Cela se traduit en général par des itérations supplémentaires dans l'algorithme.

Nous proposons dans ce paragraphe de modifier l'étape de détection globale pour tenter d'éviter ces inconvénients. Nous présentons le principe de cette modification, le nouvel algorithme correspondant et l'étude de complexité associée.

#### *Modifications de l'algorithme*

Les trois fonctions *DétectionGlobale*, *OptimLinéaire* et *OptimLocale* sont modifiées.

Dans la fonction *DétectionGlobale*, au lieu de ne détecter que la particule la plus vraisemblable, nous détectons une liste de particules candidates, triée dans l'ordre de vraisemblance. Cette détection est effectuée de la façon suivante :

1. Pour chaque couple de paramètres  $(z_p, r_p)$ , calcul du critère  $Q_k$  sur une grille de l'espace  $(x, y)$  et recherche des  $n_0$  premiers maxima locaux suivant  $(x, y)$  à  $(z_p, r_p)$  fixés. Mémoire de leurs coordonnées et de leur valeur dans une liste. La valeur  $n_0$  est choisie pour se limiter aux maxima de plus forte valeur, elle a été fixée à 20 dans notre algorithme.

2. De manière indépendante pour chaque rayon  $r_p$ , validation des maxima locaux de chaque plan  $z_p$  en vérifiant si chacun d'eux est supérieur à 2 maxima qui sont voisins selon  $(x, y)$  et situés dans les plans précédents et suivant selon  $z$ .
3. Pour chaque maximum local validé de chaque rayon  $r_p$ , validation de ce maximum s'il est supérieur à 2 maxima qui sont voisins selon  $(x, y, z)$  et dont le rayon est strictement inférieur ou supérieur.
4. Constitution de la liste finale de particules candidates en retenant tous les maxima triés par ordre de vraisemblance avec, pour chaque particule  $l$ , ses paramètres initiaux  $(x_l^0, y_l^0, z_l^0, r_l^0)$ . Le nombre de maxima de cette liste sera noté  $n_1$ . Notons qu'il peut être supérieur à  $n_0$  puisque qu'un même maximum ne se retrouve que sur un nombre limité de plans  $(z_p, r_p)$ .

Dans la fonction *OptimLinéaire*, nous déterminons par optimisation linéaire, le paramètre  $I_0^0$  et les paramètres  $\alpha_l^0$  de toutes les particules candidates. Cette optimisation est effectuée simultanément sur toutes les particules en résolvant matriciellement le système d'équation de type moindre carrés linéaires obtenu à partir du critère  $\mathcal{P}_k$  pour  $(x, y, z, r)$  fixés. A l'issue de cette optimisation, les particules ne vérifiant pas le critère  $\alpha_l^0 > \epsilon_1$  sont supprimées de la liste.

Les paramètres des  $n_2$  particules restantes sont ensuite ajustés par optimisation locale du critère  $\mathcal{P}_k$  (fonction *OptimLocale*). Cette optimisation est effectuée particule par particule à partir d'un hologramme résiduel obtenu en retirant d'une part la contribution des particules validées dans la phase d'optimisation linéaire et d'autre part la contribution des particules déjà ré-optimisées localement. Le fait d'enlever la contribution des particules validées dans *OptimLinéaire* permet de réduire le bruit et les interférences entre particules lors de l'optimisation d'une particule. A l'issue de cette optimisation, les particules ne vérifiant pas le critère  $\alpha_l^+ > \epsilon_2$  sont supprimées de la liste. Il reste alors  $n_3$  particules.

Ci-dessous le nouvel algorithme complet.

#### Algorithme 2

<b>Entrée</b> : $d[i,j]$	{hologramme numérique }
<b>Sortie</b> : $\{x_k^+, y_k^+, z_k^+, r_k^+\}_{k=1, \dots, n}$	{paramètre des particules }
$k \leftarrow 0$	
$d_0[i, j] \leftarrow d[i, j]$	
$n_3 \leftarrow 1$	
<b>Tant que</b> ( $n_3 > 0$ )	
$(x_l^0, y_l^0, z_l^0, r_l^0)_{l=1, \dots, n_1} = \text{DéetectGlobale}(d_k)$	{détection globale }
$I_0^0, (\alpha_l^0)_{l=1, \dots, n_2} = \text{OptimLinéaire}(d_k, \{1, \dots, n_2\})$	{valeurs initiales pour $I_0$ et $\alpha_l$ } {et vérification que $\alpha_l^0 > \epsilon_1$ }
$d_k = d_k - \sum_{l=0}^{n_1} m_l^0$	{soustraction des $n_1$ particules }
$n_3 \leftarrow 0$	
<b>Pour</b> ( $l=0$ à $n_2$ )	
$d_k \leftarrow d_k + m_l^0$	{ajout particule $l$ }
$(x_k^+, y_k^+, z_k^+, r_k^+, I_k^+, \alpha_k^+) = \text{OptimLocale}(d_k, l)$	{optimisation locale particule $l$ }
<b>Si</b> ( $\alpha_k^+ > \epsilon_2$ ) <b>alors</b>	{et vérification que $\alpha_k^+ > \epsilon_2$ }
$d_k \leftarrow d_k - m_l^+$	{gommage particule $l$ }
$k \leftarrow k + 1$	
$n_3 \leftarrow n_3 + 1$	
<b>FinSi</b>	
<b>Fait</b>	
<b>Fait</b>	

## Complexité

Par rapport à la version initiale de l'algorithme, l'étape de détection globale n'est pas faite à chaque itération, mais seulement lorsque les  $n_z^2$  particules initiales ont été optimisées. Le nombre  $n_z$  des particules finalement retenues lors d'une itération de la boucle *Tant que* varie d'une itération à l'autre jusqu'à devenir nul lorsque toutes les particules ont été détectées. Si on note  $n_g$  le nombre d'itérations de la boucle *Tant que*, la complexité de cette nouvelle version de l'algorithme est donnée par :

$$C = O(n_g n_{zr} N^2 \log_2(N)) + O(n_p n_{opt} N^2)$$

## 4. Evaluation des performances

Nous comparons ici les temps d'exécution obtenus pour chacune des versions de l'algorithme : la version initiale et la version modifiée.

### Protocole d'évaluation

Nous avons généré des hologrammes de synthèse comportant 20, 50 et 100 particules de  $45\mu\text{m}$  de diamètre tirées aléatoirement dans un volume de  $15,4 \times 15,4 \times 100 \text{ mm}^3$ . Le capteur est de  $1024 \times 1024$  pixels avec une taille de pixel de  $10\mu\text{m}$ . Comme ce capteur permet de couvrir les  $2/3$  de la surface projetée du volume de mesure, nous recherchons les particules dans un champ allant jusqu'à 2 fois la taille du capteur. L'hologramme est situé à une distance moyenne de  $250 \text{ mm}$  et l'enregistrement est fait avec une longueur d'onde de  $532\text{nm}$ .

Nous avons implanté l'algorithme sous Yorick/Yeti (<http://yorick.sourceforge.net>) un logiciel libre de traitement de données écrit par D. Munro/E. Thiébaud. Il était exécuté sur un PC sous Linux à base de quadricoeur Intel Xeon 3.16GHz avec 4Go de RAM (utilisation d'un seul cœur).

### Résultats

Le Tableau 1 ci-dessous présente les résultats obtenus en terme de temps de calcul pour l'algorithme initial (Algo 1) et l'algorithme modifié (Algo 2). Les temps  $t_1$  et  $t_2$  représentent les temps d'exécution moyens respectifs d'une étape de détection globale et d'une étape d'optimisation locale (incluant l'étape d'optimisation linéaire). Le temps  $t$  est le temps d'exécution global pour le dépouillement de l'hologramme complet. Lors de l'étape de détection globale, la grille d'échantillonnage  $(x,y)$  est de taille  $2N=2048$  et le nombre  $n_{zr}$  de plans  $z$  est de 100 (il n'y a qu'un seul rayon). Ces valeurs correspondent à des paramètres classiquement utilisés sur des hologrammes réels.

Algo	$N$	$n_p$	$n_g$	$n_{zr}$	$n_{opt}$	$t_1$ (s)	$t_2$ (s)	$t$ (s)	gain
Algo1	1024	17	19	100	15,5	177,56	27,82	3491	-
Algo2	1024	20	3	100	18,8	213,16	36,92	1378	2,5
Algo1	1024	51	55	100	16,9	171,67	29,68	10269	-
Algo2	1024	48	3	100	18,5	220,49	35,36	2359	4,3
Algo1	1024	100	102	100	15,4	165,23	20,68	18591	-
Algo2	1024	100	5	100	17,8	209,58	27,46	3794	4,9

Tableau 1 : temps de calcul comparé pour les 2 algorithmes

On peut constater des différences sur le nombre de particules détectées par les algorithmes. Ces différences correspondent à des fausses détections ou des particules non détectées. Le gain en temps de calcul est reporté dans la dernière colonne. Il montre que l'algorithme modifié permet de gagner un facteur de 2,5 à 4,9 sur le temps de calcul en fonction du nombre de particules sur l'hologramme. Ce gain est principalement dû au nombre d'itérations  $n_g$  de l'étape de détection grossière qui vaut 3 ou 5 avec l'algorithme 2 et environ  $n_p+2$  avec l'algorithme 1 (on s'arrête lors de 2 fausses détections consécutives). Notons que le temps de l'étape de détection grossière est plus important dans

l'algorithme 2 car on effectue en plus la détection des maxima locaux. Le temps de l'optimisation est aussi légèrement plus long à cause de la phase de suppression des particules candidates avant l'optimisation.

## 5. Conclusion

En partant de l'analyse de l'algorithme de dépouillement par optimisation proposé dans [9], nous avons présenté des modifications visant à réduire la complexité de cet algorithme, sans pour autant réduire la précision des calculs. Le principe de ces modifications est de détecter plusieurs particules lors de l'étape de détection grossière afin de réduire le temps d'exécution total de cette étape. Nous avons implanté ces modifications et nous avons évalué leur impact sur le temps d'exécution à partir d'hologrammes synthétiques.

Le gain en temps de calcul que nous avons constaté varie d'un facteur 2,5 à 4,9 suivant le nombre de particules de l'hologramme, ce qui correspond à un temps de dépouillement pouvant descendre jusqu'à environ 40s par particule. Cette amélioration est significative et permet d'étendre le champ d'utilisation pratique de cette méthode de dépouillement.

La précision de détermination des paramètres des particules devrait être aussi bonne voire meilleure que pour l'algorithme initial (l'influence des autres particules devrait être moins forte dans l'étape d'optimisation grâce à la suppression de leur contribution avant optimisation), mais ce point devra être étudié spécifiquement en perspective de ces travaux. De même, une étude sur des hologrammes bruités et sur des hologrammes réels devra être menée.

## 6. Bibliographie

1. J. GIRE, L. DENIS, C. FOURNIER, E. THIEBAUT, F. SOULEZ, C. DUCOTTET, "Digital holography of particles: benefits of the "inverse problem" approach", *Measurement Science and Technology*, vol. 19, n° 7, p. 074005, 2008.
2. K. HINSCH, *Holographic particle image velocimetry*, *Measurement Science and Technology*, vol. 13, pp. 61-72, 2002.
3. T. M. KREIS, M. ADAMS, W. JUPTNER "Methods of Digital Holography : A Comparison", *SPIE97 1997*, vol. 3098, p. 224-233, 1997.
4. T. M. KREIS, "Optical and Digital Methods", *Handbook of Holographic Interferometry*, Berlin, Wiley-VCH, 2005
5. H. MENG, G. PAN, Y. PU, S. WOODWARD, "Holographic particle image velocimetry: from film to digital recording", *Measurement Science and Technology*, Bristol: IOP Pub., c1990-, vol. 15, n° 4, pp. 673-685, 2004.
6. L. ONURAL, P. SCOTT, "Digital decoding of in-line-holograms", *Optical engineering*(Bellingham. Print), Society of Photo-Optical Instrumentation Engineers, vol. 26, n° 11, pp. 1124-1132, 1987.
7. L. ONURAL, "Diffraction from a wavelet point of view", *Optics Letters*, OSA, vol. 18, n° 11, pp. 846, 1993.
8. H. ROYER, "Holographic velocimetry of submicron particles", *Optics Communications*, vol. 20, pp. 73-75, 1977.



9. F. SOULEZ, L. DENIS, C. FOURNIER, E. THIEBAUT, C. GOEPFERT, “Inverse problem approach for particle digital holography: accurate location based on local optimisation”, *Journal of the Optical Society of America A* , vol. 24 , n° 4 , pp. 1164-1171, 2007.
10. F. SOULEZ, L. DENIS, E. THIÉBAUT, C. FOURNIER, C. GOEPFERT, “Inverse problem approach in particle digital holography: out-of-field particle detection made possible”, *Journal of Optical Society of America A*, vol. 24, n° 12, p. 3708-3716, 2007.
11. L. XU, X. PENG, Z. GUO, J. MIAO, A. ASUNDI, “Imaging analysis of digital holography”, *Optics Express*, vol. 13, n° 7, pp. 2444-2452, 2005.