



**HAL**  
open science

# Fast hologram synthesis methods for realistic 3D visualization

Antonin Gilles

► **To cite this version:**

Antonin Gilles. Fast hologram synthesis methods for realistic 3D visualization. Signal and Image processing. INSA de Rennes, 2016. English. NNT : 2016ISAR0005 . tel-01392677

**HAL Id: tel-01392677**

**<https://theses.hal.science/tel-01392677>**

Submitted on 4 Nov 2016

**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.

# Thèse

UNIVERSITE  
BRETAGNE  
LOIRE

**THESE INSA Rennes**  
*sous le sceau de l'Université Bretagne Loire*  
pour obtenir le titre de  
**DOCTEUR DE L'INSA RENNES**  
*Spécialité : Traitement du Signal et de l'Image*

présentée par

**Antonin GILLES**

**ECOLE DOCTORALE : MATISSE**

**LABORATOIRE : IETR**

## Fast hologram synthesis methods for realistic 3D visualization

**Thèse soutenue le 13/09/2016**  
devant le jury composé de :

**Pascal PICART**

Professeur, Université du Maine / *Président*

**Frédéric DUFAUX**

Directeur de Recherche CNRS, LTCI, Paris / *Rapporteur*

**Peter SCHELKENS**

Professeur, Vrije Universiteit Brussel / *Rapporteur*

**Christine GUILLEMOT**

Directrice de Recherche INRIA, Rennes / *Examinatrice*

**Laurent LUCAS**

Professeur, Université de Reims-Champagne-Ardenne / *Examineur*

**Patrick GIOIA**

Ingénieur de Recherche, Orange Labs, Rennes / *Co-encadrant*

**Rémi COZOT**

Maître de conférences HDR, Université de Rennes 1 / *Co-directeur de thèse*

**Luce MORIN**

Professeur, INSA de Rennes / *Directrice de thèse*



Fast hologram synthesis methods for realistic  
3D visualization

-

Méthodes de synthèse d'hologramme rapides  
pour la visualisation 3D réaliste

Antonin Gilles



En partenariat avec

**b com**





This work has been achieved within the Institute of Research and Technology b<>com, dedicated to digital technologies. It has been funded by the French government through the National Research Agency (ANR) Investment referenced ANR-A0-AIRT-07.



## Acknowledgements

I would like to thank my PhD supervisors, Patrick Gioia, Rémi Cozot and Luce Morin, for their guidance throughout my thesis. They helped me to be more insightful and ambitious with my research, and I could not have achieved all this work without their valuable and critical advice. I also feel grateful to my lab and project managers, Jean-Yves Aubié and Danièle Cloatre, for their confidence and support during these three years.

All my thanks to the members of my thesis committee: Frédéric Dufaux, Peter Schelkens, Christine Guillemot, Laurent Lucas and Pascal Picart, for accepting to evaluate my work. Their constructive feedback helped me improve this thesis and point out new research perspectives.

During over two months, I had the great opportunity and honor to join Kenji Yamamoto's team at the National Institute of Information and Communications Technology (NICT), in Tokyo, Japan. I would like to thank him and all the members of his team: Takanori Senoh, Ryutaro Oi, Hisayuki Sasaki, Koki Wakunami and Yuko Kameyama, for their warm welcome, their insightful advice, and for giving me access to their holographic display and wavefront printer. Our very interesting discussions during and after my stay at NICT were a rewarding experience, and I hope that our paths will cross again in the future.

Many thanks to all my co-workers and friends from b<>com, particularly Adrien, Andrey, Cambodge, Charlène, David, Fabrice, Franck, Nicolas, Pierre, Thibaud, Ulrich and Victorien for the very good times we spent together. Many thanks also to my friends from school and university with whom I spent so many good times: Eve-Pernelle, Manuel, Quentin, Thomas, Antoine, Baptiste, Clément, Guillaume, JD, Leslie, Lucas, Pauline, Pierre, Yann and Yannis. I would also like to thank my friends from Dinan and Nantes: Adeline, Antoine, Aude and Simon for making me laugh every time we meet.

Last but not least, I would like to thank my parents Yvon and Marie-Annick, my sister Hyacinthe and my girlfriend Olivia, without whom I could not have achieved this work. I cannot thank them enough for their invaluable daily support and encouragement throughout my thesis, and more generally during all my studies.



# Résumé en Français

## 1 Introduction

Avec le développement des environnements immersifs et collaboratifs, le public souhaite toujours plus de sentiment de présence pour les interactions à distance. Les systèmes de téléprésence incluent aujourd'hui la spatialisation du son et la visualisation des interlocuteurs en grandeur nature. Ainsi, la capacité à fournir une visualisation 3D avec une illusion du relief naturelle et réaliste est devenue l'une des caractéristiques les plus attendues pour ces systèmes [145].

La stéréoscopie est aujourd'hui la méthode la plus largement utilisée pour la capture et la diffusion de vidéos 3D. Ce procédé consiste à créer une illusion du relief en envoyant deux vues différentes d'une même scène à chacun des yeux de l'utilisateur. Grâce à sa simplicité d'implémentation et à sa compatibilité avec les systèmes de visualisation 2D existants, la stéréoscopie a très vite attiré l'attention de nombreux chercheurs et industriels [193]. Cependant, cette technologie présente toujours un certain nombre de contraintes et de limitations, comme la nécessité de porter des lunettes pour les écrans stéréoscopiques ou la zone de visualisation limitée des écrans auto-stéréoscopiques [32]. De plus, la stéréoscopie ne fournit pas de parallaxe de mouvement et crée un conflit entre les indices d'accommodation et de vergence, ce qui peut produire une fatigue visuelle ou des maux de tête lors d'une utilisation prolongée [56].

En réponse à ces limitations, plusieurs technologies alternatives ont été proposées ces dernières années. Parmi celles-ci, l'holographie est souvent considérée comme la plus prometteuse, puisqu'elle crée l'illusion du relief la plus naturelle et réaliste possible. En effet, l'holographie fournit tous les indices de perception de la profondeur du système visuel humain sans la nécessité de porter de lunettes et sans créer de fatigue visuelle [157]. Pour créer l'illusion de profondeur, un hologramme diffracte un faisceau de lumière incident pour lui donner la forme de l'onde lumineuse qui serait transmise ou réfléchi par une scène donnée. Ainsi, l'utilisateur peut percevoir la scène comme si elle était physiquement présente devant lui.

L'acquisition optique d'un hologramme s'effectue à l'aide d'un processus d'interférence entre deux faisceaux lumineux issus d'un laser. Le système optique doit être extrêmement stable car une vibration de l'ordre de la longueur d'onde du laser utilisé peut modifier radicalement les franges d'interférence. Cette contrainte, ainsi que la durée d'exposition et de développement du support photosensible, rend l'acquisition optique de vidéos holographiques très difficile [173]. En outre l'holographie optique conventionnelle suppose que la scène est éclairée par un laser, ce qui interdit l'acquisition directe de scènes illuminées par un éclairage naturel.

Pour éviter ces contraintes, plusieurs méthodes ont été proposées pour remplacer le processus d'acquisition optique d'un hologramme par des calculs numériques. Grâce à ces méthodes, il est possible de générer l'hologramme d'une scène synthétique ou réelle sans passer par le processus physique d'interférence entre deux ondes lumineuses. Les principales limitations liées à l'holographie optique conventionnelle peuvent ainsi être évitées. Ce travail se situe dans ce contexte et vise à relever les défis rencontrés par les chercheurs dans le domaine de la génération numérique d'hologrammes.

## 2 État de l'art des méthodes de synthèse d'hologrammes

Dans cette section, nous présentons les différentes approches utilisées par les méthodes de synthèse d'hologrammes de l'état de l'art. Ces méthodes diffèrent suivant le type de données d'entrée utilisées pour représenter la géométrie de la scène: certaines utilisent des modèles 3D, d'autres des données multi-vues.

### 2.1 A partir d'un modèle 3D

Pour calculer l'hologramme d'une scène à partir de son modèle 3D, les méthodes de l'état de l'art échantillonnent généralement cette scène par un ensemble de primitives. L'hologramme est alors calculé comme la somme des ondes lumineuses émises par chacune des primitives de la scène. Deux types de primitives sont communément utilisées: les points (approche *point-source*) et les polygones (approche *wave-field*).

Dans l'approche *point-source*, les scènes 3D sont représentées sous la forme d'un nuage de points lumineux non maillés. L'hologramme est alors calculé comme la somme des ondes sphériques émises par ces points [15]. Les occultations dans la scène sont généralement prises en compte à l'aide d'un test de visibilité pour s'assurer de la présence ou non d'obstacles entre chaque point de la scène et chaque pixel de l'hologramme [185]. Cette approche est très simple à mettre en œuvre et n'impose aucune restriction sur la géométrie de la scène.

De plus, les techniques de synthèse d'image pour le calcul de l'illumination de la scène, telles que l'ombrage de Phong [139], peuvent être directement intégrées lors du calcul de l'hologramme pour produire un rendu photo-réaliste de la scène [91, 223]. Cependant, la complexité algorithmique de cette approche est très importante, puisqu'elle nécessite un calcul par point de la scène par pixel de l'hologramme.

Dans l'approche *wave-field*, les scènes 3D sont représentées sous la forme d'un maillage de polygones. L'hologramme est alors calculé comme la somme des ondes lumineuses émises par chacun des polygones de la scène [99]. L'onde lumineuse émise par un polygone est calculée en utilisant des formules de propagation de la lumière, qui s'expriment sous la forme d'un produit de convolution [53]. Leur calcul nécessite donc l'utilisation de trois Transformées de Fourier, et prend donc plus de temps que le calcul de l'onde sphérique émise par un point. Cependant, les ondes lumineuses émises par l'ensemble des points d'un même polygone sont calculées en une seule fois en utilisant les formules de propagation de la lumière. L'approche *wave-field* est donc plus efficace que l'approche *point-source* lorsque la scène comporte de grandes surfaces planes comportant un grand nombre de points. En revanche, lorsque la scène contient des formes à la géométrie complexe, un grand nombre de polygones contenant peu de points est nécessaire pour l'échantillonner, et l'approche *point-source* est donc plus efficace. Comme pour l'approche *point-source*, les techniques de synthèse d'image pour le calcul de l'illumination de la scène peuvent être intégrées lors du calcul de l'hologramme [60, 125, 151, 199].

Ces deux approches présentent l'avantage de produire des hologrammes de scènes réalistes avec tous les indices de perception de la profondeur du système visuel humain. Cependant, pour prendre en compte l'illumination de la scène, la géométrie, les matériaux ainsi que la position des sources lumineuses doivent être modélisés avec précision. Ainsi, ces méthodes ne peuvent pas trivialement être appliquées à la génération d'hologrammes de scènes réelles.

## 2.2 A partir de données multi-vues

Pour calculer l'hologramme d'une scène réelle, il est possible d'utiliser des données multi-vues. Deux approches sont couramment utilisés pour la synthèse d'hologrammes à partir de ce type de données: la projection multi-vue et les stéréogrammes holographiques.

La première étape des méthodes basées sur la projection multi-vue est d'acquérir des projections de la scène depuis différents points de vue, en utilisant par exemple une matrice de caméras [92]. Le nombre de vues à acquérir est égal à la résolution de l'hologramme. Chaque pixel de l'hologramme est alors calculé en multipliant l'image correspondante par une fonction d'étalement du point (*Point Spread Function* ou *PSF* en anglais) puis

par sommation du résultat [2, 100]. Comme les occultations ainsi que l'illumination de la scène sont naturellement enregistrées dans les images acquises, il n'est pas nécessaire d'effectuer de calcul supplémentaire pour les inclure dans l'hologramme à générer. De plus, les hologrammes calculés par cette approche fournissent tous les indices de perception de la profondeur du système visuel humain. Cependant, étant donné que chaque image correspond à un seul pixel de l'hologramme, il est nécessaire d'acquérir une très grande quantité de vues de la scène, en particulier pour les hologrammes de grande résolution.

Plusieurs méthodes ont été proposées pour réduire le nombre de vues à acquérir. Dans [100, 161], les auteurs proposent de calculer un hologramme à parallaxe horizontale en ne faisant l'acquisition des images que le long de l'axe horizontal. Même si cette méthode permet de réduire considérablement le nombre de projections nécessaires, l'hologramme ainsi calculé est incapable de fournir la parallaxe verticale, limitant fortement la perception du relief. De plus, cette méthode nécessite toujours une vue par colonne de pixels de l'hologramme, ce qui reste difficile à acquérir pour les hologrammes de grande résolution. Dans [76], les auteurs proposent de ne faire l'acquisition que de quelques images le long de l'axe horizontal et de synthétiser des vues intermédiaires en utilisant une méthode d'interpolation, réduisant ainsi le nombre d'images au prix d'une plus grande complexité algorithmique. Enfin, il est également possible d'utiliser un réseau lenticulaire afin de capturer simultanément un grand nombre de projections [120, 163]. Grâce à cette méthode, il est possible de n'utiliser qu'une seule caméra pour capturer les données multi-vues, sans qu'il soit nécessaire de la déplacer mécaniquement. Le nombre de projections acquises est donné par le nombre de micro-lentilles du réseau lenticulaire, et la résolution de chaque image élémentaire est donnée par la résolution de la caméra divisée par le nombre de micro-lentilles. Par conséquent, ce processus d'acquisition conduit à un compromis: plus le réseau lenticulaire contient de micro-lentilles, plus le nombre de projections acquises sera important, mais plus leur résolution sera faible, et inversement.

Les stéréogrammes holographiques sont un autre type d'hologrammes qui peuvent être calculés à partir de données multi-vues [210]. Dans cette approche, l'hologramme est spatialement divisé en plusieurs éléments holographiques, appelés hogels. Les hogels ont pour rôle de distribuer les projections 2D de la scène vers différentes directions d'observation. Cette approche présente une faible complexité algorithmique et ne nécessite pas autant d'images que les méthodes utilisant la projection multi-vues. En outre, comme pour les méthodes par projection multi-vue, il n'est pas nécessaire d'effectuer de calcul supplémentaire pour inclure les occultations ainsi que l'illumination de la scène dans l'hologramme à générer. Cependant, étant donné que chaque hogel ne correspond qu'à une seule projection 2D de la scène, l'indice d'accommodation est sacrifié, ce qui limite fortement la perception du relief.

---

Plusieurs méthodes ont été proposées pour améliorer l'illusion de profondeur fournie par les stéréogrammes holographiques [9, 74, 174, 203]. Cependant, ces méthodes nécessitent une description précise de la géométrie de la scène. Par conséquent, elles ne peuvent être utilisées que pour des modèles 3D synthétiques ou pour des données multivue-plus-profondeur.

### 2.3 Limitations et verrous technologiques

Grâce à ses nombreux atouts, l'holographie pourrait remplacer la stéréoscopie dans les systèmes de visualisation 3D. Toutefois, les méthodes de génération numérique d'hologrammes doivent répondre à plusieurs exigences. Tout d'abord, ces méthodes doivent créer des hologrammes qui fournissent tous les indices de perception de la profondeur du système visuel humain afin de créer une illusion du relief naturelle et réaliste. Deuxièmement, le rendu de la scène à afficher doit être aussi réaliste que possible, ce qui signifie que les occultations ainsi que l'illumination de la scène doivent être reproduits fidèlement. Enfin, afin d'avoir application dans le domaine de la visioconférence ou de la téléprésence, les méthodes de calcul doivent être en mesure de produire des hologrammes de scènes réelles en temps réel.

Cependant, étant donné qu'un hologramme créé l'illusion de la profondeur en diffractant le faisceau de lumière incident, sa zone de visualisation dépend de la taille de ses pixels. De plus, celle-ci doit être de l'ordre de grandeur de la longueur d'onde de la lumière visible, c'est-à-dire de quelques centaines de nanomètres. Ainsi, un hologramme en couleur de grande taille avec une grande zone de visualisation nécessite une énorme quantité d'informations. Par exemple, un hologramme rouge monochromatique (pour une longueur d'onde de 640nm) de taille (10cm × 10cm) avec une zone de visualisation de (120° × 120°) nécessite une taille de pixel de 369.5nm et une résolution de plus de (270K × 270K). Le même hologramme en couleur nécessite au moins trois fois plus d'informations.

En raison de cette énorme quantité de données à traiter, il est encore très difficile pour les méthodes de l'état de l'art de répondre à toutes les exigences mentionnées plus haut. Ainsi, la plupart des méthodes de génération d'hologrammes font un compromis entre le réalisme de la scène à afficher, l'illusion du relief et le temps de calcul. Par exemple, pour accélérer le calcul, certaines méthodes font le choix de créer des hologrammes qui ne fournissent pas tous les indices de perception de la profondeur ou utilisent un rendu de la scène simplifié. L'objectif de cette thèse est donc de proposer de nouveaux algorithmes capables de produire des hologrammes de scènes réalistes avec une forte illusion de la profondeur et une complexité de calcul réduite. Les contributions de cette thèse sont organisées en deux parties. Nous avons tout d'abord cherché à diminuer le temps de calcul d'un hologramme afin de se rapprocher du temps réel, puis nous avons travaillé sur l'amélioration de la qualité visuelle des scènes.

### 3 Accélération du calcul de l'hologramme

Dans la première partie de ce travail, nous avons proposé deux algorithmes permettant de se rapprocher du temps réel. Premièrement, nous avons développé une méthode hybride combinant deux approches complémentaires: les approches *point-source* et *wave-field*. Deuxièmement, nous avons cherché à accélérer cette méthode en supprimant les redondances temporelles entre les trames successives d'une vidéo 3D.

#### 3.1 Méthode hybride *point-source/wave-field*

Comme mentionné plus haut, les approches *point-source* et *wave-field* sont complémentaires l'une de l'autre. En effet, alors que l'approche *wave-field* est très efficace lorsque la scène contient de grandes surfaces planes, telles que des routes ou des bâtiments, l'approche *point-source* est plus efficace lorsque la scène contient des formes à la géométrie complexe, telles que des arbres ou des corps humains. Cependant, la plupart des scènes réelles contiennent à la fois des formes à la géométrie complexe et des surfaces planes. Par conséquent, ces deux approches ne sont jamais totalement efficaces pour calculer l'hologramme d'une scène réelle.

Pour pallier à ces limitations, nous proposons une nouvelle méthode de génération numérique d'hologrammes basée sur une approche hybride *point-source/wave-field*. Alors que la plupart des méthodes de l'état de l'art ont pour objectif de diminuer la complexité de l'une ou l'autre de ces approches indépendamment, notre méthode les combine et tire ainsi parti des avantages de chacune d'entre elles.

L'algorithme proposé comporte trois étapes. Tout d'abord, la scène 3D est uniformément subdivisée en un certain nombre de plans de profondeur parallèles au plan de l'hologramme. Le nombre de plans est défini de telle sorte que la séparation entre deux plans reste invisible pour l'observateur. Ensuite, les plans de la scène sont classés en deux catégories: les plans *wave-field* et les plans *point-source*. Cette classification est effectuée en fonction du nombre de points qu'ils contiennent : si le nombre de points contenus dans le plan dépasse une certaine valeur seuil, alors celui-ci est considéré comme étant un plan *wave-field*, sinon il est considéré comme un plan *point-source*. Enfin, les ondes lumineuses émises par chacun des plans sont propagées dans le plan de l'hologramme en utilisant une approche *wave-field* ou une approche *point-source* en fonction de la catégorie du plan, et sont additionnées pour obtenir l'hologramme final.

Les résultats expérimentaux montrent que cette combinaison d'approches est toujours plus efficace que les approches *point-source* et *wave-field*, quelle que soit la configuration de la scène. Ainsi, le temps de calcul de l'hologramme a été réduit de plus de 65% par rapport aux méthodes *point-source* et *wave-field* conventionnelles.

## 3.2 Suppression des redondances temporelles

Bien que la méthode hybride surpasse les approches *point-source* et *wave-field* conventionnelles en termes de temps de calcul, elle n'est capable de calculer que 5 hologrammes couleur de résolution ( $2048 \times 2048$ ) par seconde, pour une scène contenant 262144 points, ce qui n'est pas suffisant pour un système temps-réel. Afin de réduire davantage le temps de calcul de l'hologramme, il est possible de tirer parti des redondances temporelles d'une vidéo 3D.

En effet, il n'existe que peu de changements entre les trames successives d'une vidéo 3D, et l'hologramme n'a pas besoin d'être recalculé entièrement pour chaque nouvelle trame. Plusieurs méthodes de l'état de l'art tirent parti de ces redondances [33, 35, 94, 142]. Le principe de fonctionnement de ces méthodes est de détecter les changements dans la scène et de ne mettre à jour que l'hologramme des points ayant changé. Comme seules de petites régions de l'hologramme sont mises-à-jour à chaque nouvelle trame, ces méthodes permettent de réduire considérablement la charge de calcul. Cependant, ces méthodes de l'état de l'art fonctionnent toutes sur une approche *point-source*, et l'exploitation des redondances temporelles dans le cas de l'approche hybride n'a jamais été étudiée.

Ainsi, nous proposons une nouvelle méthode pour le calcul de vidéos holographiques couleur basée sur l'approche hybride en prenant en compte ces redondances temporelles. La méthode prend comme données d'entrée une vidéo 3D dont chaque trame comporte une image d'intensité et une carte de profondeur. Pour chaque trame de la vidéo, un nuage de points est extrait à partir des données de profondeur, chaque point ayant une amplitude initiale proportionnelle à la valeur du pixel correspondant dans l'image d'intensité. Comme chaque carte de profondeur est encodée comme une image en niveaux de gris 8-bits, le nuage de points extrait est naturellement subdivisé en 256 plans de profondeur parallèles au plan de l'hologramme.

Pour la première trame, l'hologramme  $H_0$  est calculé à partir de l'image d'intensité et de la carte de profondeur en utilisant la méthode hybride présentée plus haut. Pour les trames suivantes, les données d'intensité et de profondeur sont d'abord comparées à celles de la trame précédente afin d'en extraire les différences. Un critère est défini pour déterminer si ces changements correspondent à une grande partie de la scène ou non. Si les changements dans la scène sont importants, l'hologramme  $H_i$  est calculé en utilisant la méthode hybride originale. Sinon, la méthode ne calcule l'hologramme que des points ayant changé et l'ajoute à l'hologramme  $H_{i-1}$  de la trame précédente pour obtenir  $H_i$ .

Les résultats expérimentaux montrent que dans le cas où il n'existe que peu de changements entre les trames successives d'une vidéo 3D, le temps de calcul d'un hologramme couleur de résolution ( $2048 \times 2048$ ) peut être réduit à 16,60ms. En d'autres termes, la méthode proposée est capable de calculer des hologrammes couleur en temps-réel, à 60

trames par seconde. De plus, dans le cas où une importante part de la scène change entre deux trames successives, la méthode proposée est toujours plus efficace que les méthodes *wave-field* et *point-source* conventionnelles.

## 4 Amélioration de la qualité visuelle

Dans la deuxième partie de ce travail, nous avons proposé deux algorithmes afin d'améliorer la qualité visuelle des scènes. Tout d'abord, nous avons perfectionné la méthode hybride pour tenir compte des occultations dans la scène. Enfin, nous avons proposé une méthode pour le calcul d'un hologramme à partir de données multivue-plus-profondeur avec prise en compte des reflets spéculaires.

### 4.1 Prise en compte des occultations

La méthode hybride est toujours plus efficace que les approches *point-source* et *wave-field* conventionnelles en termes de temps de calcul. De plus, lorsqu'elle est combinée à un algorithme de suppression des redondances temporelles, cette méthode est capable de produire des vidéo holographiques couleur en temps réel. Cependant, la méthode hybride originale ne prend pas en compte les occultations dans la scène, limitant fortement le réalisme de la scène et l'illusion du relief. En effet, les occultations forment l'un des indices de perception de la profondeur le plus important pour le système visuel humain.

En fait, dans la méthode hybride originale et la méthode de suppression des redondances temporelles, les occultations sont naturellement enregistrées dans les images d'intensité et cartes de profondeur. Malheureusement, cette prise en compte des occultations n'est valable que pour de petits angles de visualisation. En effet, contrairement aux images synthétiques 2D, les hologrammes fournissent la parallaxe de mouvement, qui est couplée aux occultations. Ainsi, la prise en compte des occultations en holographie implique que la visibilité des objets change en fonction de la position de l'observateur.

Pour pallier à cette limitation, nous proposons une méthode efficace pour le calcul de l'occultation d'une onde lumineuse par un point, permettant aux plans *point-source* d'occulter correctement les plans *wave-field*, et vice-versa. Dans cette méthode, la lumière émise par la scène est propagée et occultée d'un plan à un autre en partant du plan le plus lointain du plan de l'hologramme. Pour chaque plan, les opérations de propagation et d'occultation sont calculées en utilisant soit une approche *point-source*, soit une approche *wave-field*, en fonction du nombre de points qu'il contient. Enfin, l'algorithme calcule la propagation du dernier plan de la scène jusqu'au plan de l'hologramme pour obtenir l'hologramme final.

Les résultats expérimentaux montrent que cette méthode reproduit fidèlement les occultations dans la scène sans augmenter de manière significative le temps de calcul de la méthode hybride originale. De plus, le temps de calcul de cette méthode a été réduit de plus de 45% comparé aux méthodes *point-source* et *wave-field* conventionnelles.

## 4.2 Prise en compte des reflets spéculaires

Comme indiqué plus haut, il est encore très difficile pour les méthodes de l'état de l'art de calculer l'hologramme d'une scène réelle permettant de fournir tous les indices de perception de la profondeur et une reproduction précise de l'illumination avec seulement quelques projections de la scène. En effet, même si les approches *point-source* et *wave-field* reproduisent tous les indices de perception de la profondeur, elles nécessitent une description précise de la géométrie, des matériaux et des sources lumineuses de la scène pour prendre en compte son illumination. Ainsi, ces méthodes ne peuvent prendre en compte les reflets spéculaires que pour des scènes purement synthétiques. A l'inverse, dans le cas des méthodes utilisant des données multi-vues, comme les occultations ainsi que l'illumination sont naturellement enregistrées dans les images acquises de la scène, il n'est pas nécessaire d'effectuer de calcul supplémentaire pour les inclure dans l'hologramme à générer. Cependant, étant donné que chaque image correspond à un seul pixel de l'hologramme, il est nécessaire d'acquérir une très grande quantité de vues de la scène pour calculer l'hologramme avec une méthode par projection multi-vue. Les stéréogrammes holographiques, quant à eux, ne nécessitent pas autant d'images que les méthodes utilisant la projection multi-vues, mais sont incapables de produire l'indice d'accommodation, limitant fortement l'illusion du relief.

Afin de contourner ces limitations, nous proposons une nouvelle méthode pour le calcul d'un hologramme à partir de données multivue-plus-profondeur avec prise en compte des reflets spéculaires. L'algorithme proposé comporte trois étapes. Tout d'abord, des projections 2D+Z de la scène sont acquises depuis différents points de vue. Ensuite, la géométrie de la scène est reconstruite à partir de ces données sous la forme d'un nuage de points. Cette étape de reconstruction permet de n'utiliser qu'un nombre restreint de projections de la scène sans sacrifier aucun indice de perception de la profondeur. De plus, afin de prendre en compte les reflets spéculaires, chaque point de la scène est considéré comme ayant une émission différente suivant la direction d'observation. Enfin, la lumière émise par la scène est propagée jusqu'au plan de l'hologramme afin d'obtenir l'hologramme final.

Les résultats expérimentaux montrent que cette méthode est capable de fournir tous les indices de perception de la profondeur avec seulement quelques projections de la scène. De plus, les reflets spéculaires sont fidèlement reproduits avec une charge de calcul réduite.

## 5 Conclusion

Au cours de cette thèse nous avons étudié deux algorithmes pour réduire le temps de calcul de l'hologramme. Le premier est basé sur une approche hybride, tandis que le second tire parti des redondances temporelles pour accélérer le calcul d'hologrammes vidéo. Ces méthodes nous ont permis de générer des hologrammes couleur en temps réel. Dans la seconde partie de cette thèse, nous avons apporté des modifications à ces algorithmes pour améliorer le réalisme et la qualité visuelle des scènes sans augmenter significativement le temps de calcul de l'hologramme. Nous avons ainsi proposé deux nouvelles méthodes pour tenir compte des occultations et des reflets spéculaires de manière efficace. Ainsi, nous avons été en mesure de produire des hologrammes de scènes réalistes permettant de fournir tous les indices de perception de la profondeur avec une complexité de calcul réduite. En outre, ces travaux ont été valorisés par plusieurs publications et un brevet, dont la liste est donnée en Annexe A. Cependant, nos contributions ont certaines limites qui feront l'objet de travaux futurs.

Tout d'abord, la méthode hybride est limitée à une modélisation en tranches de la scène, ce qui n'est pas la représentation la plus réaliste. Afin de contourner cette limitation, cette approche devrait être généralisée à une modélisation en maillage de polygones. Dans ce cas, les polygones qui composent la scène peuvent être classés en deux catégories: les polygones *point-source* et les polygones *wave-field*. Cette classification peut être réalisée en fonction de la taille et de l'orientation de chaque polygone.

Deuxièmement, la méthode basée sur la suppression des redondances temporelles entre les trames d'une vidéo n'a été testée qu'avec des scènes entièrement synthétiques. Dans ce cas, les cartes de profondeur sont parfaites en termes de précision et d'alignement avec l'image d'intensité correspondante, et sont temporellement cohérentes d'une trame à une autre, ce qui permet l'élimination efficace des redondances temporelles. Cependant, lors de la capture d'une scène réelle avec une caméra 2D+Z, les cartes de profondeur seront inévitablement bruitées. En conséquence, la sensibilité et l'efficacité de cette méthode vis-à-vis de ces erreurs devront être étudiées. En outre, d'autres algorithmes de compression des redondances temporelles, telles que les méthodes basées sur une estimation de mouvement, doivent être évaluées pour réduire davantage le temps de calcul de l'hologramme.

Enfin, d'autres investigations sont nécessaires pour améliorer l'algorithme de prise en compte des reflets spéculaires à partir de données multivue-plus-profondeur. En particulier, l'influence du coefficient de réflexion des points sur le réalisme et la qualité visuelle de la scène doit être soigneusement évalué. Des formules doivent être dérivées afin de déterminer automatiquement la valeur de ce paramètre en fonction des caractéristiques et de la disposition des caméras utilisées. Enfin, cette méthode doit également être évaluée en utilisant des scènes réelles capturées à la lumière naturelle.

# Table of contents

<b>List of figures</b>	<b>xxiii</b>
<b>List of tables</b>	<b>xxv</b>
<b>Nomenclature</b>	<b>xxvii</b>
<b>I Introduction</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Preamble . . . . .	3
1.2 Objectives and motivations . . . . .	4
1.3 Contributions . . . . .	5
1.4 Thesis organization . . . . .	6
<b>II Background</b>	<b>9</b>
<b>2 Background</b>	<b>11</b>
2.1 Objectives and requirements of 3D video systems . . . . .	11
2.1.1 Human Visual System depth cues . . . . .	11
2.1.2 The stereoscopic technique . . . . .	13
2.1.3 Alternative technologies . . . . .	14
2.2 Conventional Optical Holography . . . . .	15
2.2.1 Optical hologram acquisition . . . . .	15
2.2.2 Optical hologram reconstruction . . . . .	17
2.2.3 Constraints and limitations . . . . .	17
2.3 Computer-Generated Holography . . . . .	18
2.3.1 3D content creation . . . . .	18

2.3.2	Object wave computation . . . . .	19
2.3.3	Hologram encoding . . . . .	19
2.3.4	3D image reconstruction . . . . .	20
2.4	Scalar diffraction theory . . . . .	21
2.4.1	The Helmholtz equation . . . . .	22
2.4.2	Rayleigh-Sommerfeld propagation . . . . .	22
2.4.3	Angular Spectrum propagation . . . . .	23
2.4.4	The Fresnel approximation . . . . .	23
2.4.5	The Fraunhofer approximation . . . . .	24
2.5	Conclusion . . . . .	25
<b>3</b>	<b>State of the art of hologram synthesis methods</b>	<b>27</b>
3.1	Hologram synthesis parameters . . . . .	27
3.1.1	Scene parameters . . . . .	27
3.1.2	Rendering parameters . . . . .	28
3.1.3	Hologram parameters . . . . .	29
3.1.4	Comparison of hologram synthesis methods . . . . .	29
3.2	Object wave computation . . . . .	30
3.2.1	Point-source approach . . . . .	30
3.2.2	Wave-field approach . . . . .	36
3.2.3	Object wave computation from multiview data . . . . .	42
3.2.4	Comparison of object wave computation approaches . . . . .	48
3.3	Hologram representation . . . . .	49
3.3.1	Amplitude holograms . . . . .	49
3.3.2	Phase holograms . . . . .	51
3.3.3	Complex modulation holograms . . . . .	51
3.4	Conclusion . . . . .	52
<b>III</b>	<b>Hologram calculation time reduction</b>	<b>55</b>
<b>4</b>	<b>Fast hologram computation using a hybrid point-source/wave-field approach</b>	<b>57</b>
4.1	Proposed method . . . . .	58
4.1.1	Overview . . . . .	58
4.1.2	Wave-field layers . . . . .	59
4.1.3	Point-source layers . . . . .	60
4.1.4	Layers classification . . . . .	61

---

4.2	Graphics Processing Unit implementation . . . . .	62
4.2.1	CUDA thread organization and memory model . . . . .	62
4.2.2	Wave-field layers . . . . .	63
4.2.3	Point-source layers . . . . .	64
4.3	Experimental results and discussion . . . . .	66
4.3.1	Input 3D scenes . . . . .	66
4.3.2	Analysis of the visual quality of the reconstructed images . . . . .	67
4.3.3	Analysis of the hologram calculation time . . . . .	69
4.4	Conclusion . . . . .	73
<b>5</b>	<b>Real-time hologram computation using temporal redundancy removal</b>	<b>75</b>
5.1	Proposed method . . . . .	76
5.1.1	Overview . . . . .	76
5.1.2	Incremental update . . . . .	76
5.2	Experimental results and discussion . . . . .	78
5.2.1	Input 3D video . . . . .	78
5.2.2	Analysis of the visual quality of the reconstructed images . . . . .	80
5.2.3	Analysis of the hologram calculation time . . . . .	82
5.3	Conclusion . . . . .	86
<b>IV</b>	<b>Hologram visual quality improvement</b>	<b>87</b>
<b>6</b>	<b>Hologram with occlusion effect using the hybrid approach</b>	<b>89</b>
6.1	Proposed method . . . . .	90
6.1.1	Case 1: Light shielding by wave-field layers . . . . .	90
6.1.2	Case 2: One point-source layer between two wave-field layers . . . . .	91
6.1.3	Case 3: Several point-source layers between each two wave-field layers . . . . .	95
6.1.4	Layers classification . . . . .	96
6.2	Graphics Processing Unit implementation . . . . .	97
6.2.1	Point-source layers . . . . .	98
6.2.2	Wave-field layers . . . . .	102
6.3	Experimental results and discussion . . . . .	102
6.3.1	Input 3D scenes . . . . .	103
6.3.2	Analysis of the visual quality of the reconstructed images . . . . .	104
6.3.3	Analysis of the hologram calculation time . . . . .	105
6.4	Conclusion . . . . .	107

<b>7</b>	<b>Hologram synthesis from MVD data with specular reflections</b>	<b>109</b>
7.1	Hologram synthesis for a diffuse scene . . . . .	110
7.1.1	Step 1: Multiview-plus-Depth capture . . . . .	110
7.1.2	Step 2: Scene geometry reconstruction . . . . .	111
7.1.3	Step 3: Light propagation and hologram encoding . . . . .	112
7.2	Rendering specular reflections . . . . .	112
7.2.1	Role of the initial phase distribution . . . . .	113
7.2.2	Creation of a new phase distribution . . . . .	114
7.2.3	Overall algorithm . . . . .	115
7.3	Experimental results and discussion . . . . .	116
7.3.1	Input 3D scenes . . . . .	116
7.3.2	Analysis of the visual quality of the reconstructed images . . . . .	117
7.3.3	Analysis of the hologram calculation time . . . . .	118
7.4	Conclusion . . . . .	120
<b>V</b>	<b>Conclusion</b>	<b>121</b>
<b>8</b>	<b>Conclusions and future work</b>	<b>123</b>
8.1	Achieved work . . . . .	123
8.2	Future work . . . . .	125
8.3	Open issues and prospects . . . . .	126
<b>VI</b>	<b>References and Appendix</b>	<b>129</b>
	<b>References</b>	<b>131</b>
	<b>Appendix A Publications and patent</b>	<b>149</b>
A.1	Scientific journals . . . . .	149
A.2	International conferences . . . . .	150
A.3	Patent . . . . .	151

# List of figures

2.1	Overview and classification of human visual system depth cues. . . . .	12
2.2	Accommodation/vergence conflict in stereoscopic systems. . . . .	13
2.3	Optical hologram recording set-up. . . . .	15
2.4	Optical hologram reconstruction set-up. . . . .	16
2.5	Full Computer-Generated Holography processing chain. . . . .	18
2.6	Wavefront printer set-up. . . . .	21
3.1	Computation of the object wave using the point-source approach. . . . .	30
3.2	Region of contribution of a given scene point. . . . .	31
3.3	Phong shading using the point-source approach. . . . .	32
3.4	Occlusion processing using an object-oriented approach. . . . .	33
3.5	Occlusion processing using an image-oriented approach. . . . .	34
3.6	Wavefront recording plane technique. . . . .	35
3.7	Wave-field approach based on a layered model of the scene. . . . .	37
3.8	Wave-field approach based on a polygonal modeling of the scene. . . . .	38
3.9	Exact occlusion processing using per-polygon tilted binary masks. . . . .	39
3.10	Simplified occlusion processing using per-polygon silhouette binary masks. . . . .	40
3.11	Simplified occlusion processing using per-object silhouette binary masks. . . . .	41
3.12	Multiview capture using several cameras arranged along the horizontal axis. . . . .	43
3.13	Object wave computation using the multiple-viewpoint-projection approach. . . . .	44
3.14	Object wave computation using the holographic stereogram approach. . . . .	46
3.15	Phase hologram computation using the Gerchberg-Saxton algorithm. . . . .	50
4.1	Scene geometry and coordinate system used by the proposed method. . . . .	58
4.2	CUDA thread organization and memory model. . . . .	62
4.3	Software structure for calculation of waves scattered by wave-field layers. . . . .	63
4.4	Software structure for calculation of waves scattered by point-source layers. . . . .	64
4.5	Test scenes used for the experiments. . . . .	66

4.6	Numerical reconstructions of the CGHs of scene <i>Ballet</i> . . . . .	68
4.7	Numerical reconstructions of the CGHs of scene <i>Dices</i> . . . . .	69
4.8	CGH calculation time depending on the number of scene points. . . . .	70
4.9	CGH calculation time depending on the number of pixels. . . . .	71
4.10	CGH calculation time depending on the distance of the scene. . . . .	72
5.1	Block-diagram of the proposed method. . . . .	77
5.2	Views and associated depth maps for nine input 3D video frames. . . . .	79
5.3	Scene points whose complex waves are updated at each frame. . . . .	80
5.4	Numerical reconstructions of nine different CGH frames. . . . .	81
5.5	Number of points whose complex waves are calculated for each frame. . . . .	83
5.6	Hologram calculation time for each frame. . . . .	84
6.1	Light shielding procedure. . . . .	90
6.2	Light shielding using the binary mask function. . . . .	91
6.3	Light shielding using the binary aperture function. . . . .	93
6.4	Block-diagram of the proposed method. . . . .	94
6.5	Software structure for the calculation of $\hat{u}_d$ when $d$ is a point-source layer. . . . .	98
6.6	Software structure for the calculation of $u_d$ when $d$ is a wave-field layer. . . . .	100
6.7	Computer graphics images of the three test scenes used for the experiments. . . . .	103
6.8	Number of scene points per layer for the three scenes. . . . .	103
6.9	Number and percentage of scene points processed by the proposed method using the point-source and wave-field approaches. . . . .	104
6.10	Numerical reconstructions of the CGHs for the three input scenes. . . . .	105
6.11	Optical reconstructions of scene <i>Dice2</i> focused on different objects. . . . .	106
6.12	Optical reconstructions of scene <i>Dice2</i> captured from different viewpoints. . . . .	106
7.1	Multiview-plus-Depth capture geometry. . . . .	110
7.2	Relation between the spectrum of the original diffuser and diffuse reflection. . . . .	113
7.3	Relation between the spectrum of the specular diffuser and specular reflection. . . . .	114
7.4	Relation between the pixels coordinates of a point and the direction from which its intensity has been captured by the camera. . . . .	116
7.5	Comparison between the numerical reconstructions and CG images. . . . .	118
7.6	Numerical reconstructions focused on different depths. . . . .	119
7.7	CGH calculation time depending on the number of MVD projections. . . . .	119

# List of tables

3.1	Comparison of object wave computation approaches . . . . .	48
4.1	Hologram synthesis parameters used for the experiments. . . . .	67
5.1	Hologram synthesis parameters used for the experiments. . . . .	79
5.2	Average number of calculated points and CGH calculation time per frame. .	85
6.1	Hologram synthesis parameters used for the experiments. . . . .	102
6.2	Comparison of the hologram computation times. . . . .	107
7.1	Hologram synthesis parameters used for the experiments. . . . .	117



# Nomenclature

## Acronyms / Abbreviations

1D	One-dimensional
2D	Two-dimensional
2DTV	2D Television
3D	Three-dimensional
3DTV	3D Television
AOM	Acousto-Optic Modulator
AR	Augmented Reality
CG	Computer Graphics
CGH	Computer-Generated Hologram
DBS	Direct Binary Search
DIBR	Depth-Image-Based Rendering
DMD	Digital Micro-mirror Display
DPCM	Differential Pulse Code Modulation
FFT	Fast Fourier Transform
GA	Genetic Algorithm
GPU	Graphics Processing Unit
GS	Gerchberg-Saxton

HMD	Head-Mounted Display
HPO	Horizontal Parallax Only
HSRM	Hidden Surface Removal Method
HVS	Human Visual System
IFTA	Iterative Fourier Transform Algorithm
IIT	Iterative Interlacing Technique
JPEG	Joint Photographic Experts Group
LCoS	Liquid Crystal on Silicon
LUT	Look-up Table
MVD	Multiview-plus-Depth
NICT	National Institute of Information and Communications Technology
PSF	Point Spread Function
PSNR	Peak Signal-to-Noise Ratio
RLE	Run-Length Encoding
RS	Ray-Sampling
SA	Simulated Annealing
SFS	Shape-From-Silhouette
SLM	Spatial Light Modulator
SSIM	Structural Similarity Index
VR	Virtual Reality
WRP	Wavefront Recording Plane

**Mathematical Symbols**

$\mathcal{F}$	Fourier transform
$j$	Imaginary unit
$k$	Wave number
$\lambda$	Wavelength of light
$\nabla^2$	Laplace operator
$\mathcal{P}^{AS}$	Angular Spectrum propagation
$\mathcal{P}^{Fh}$	Fraunhofer propagation
$\mathcal{P}^{Fr1}$	Fresnel propagation (convolution form)
$\mathcal{P}^{Fr2}$	Fresnel propagation (Fourier form)
$\mathcal{P}^{PS}$	Light propagation using the point-source approach
$\mathcal{P}^{RS}$	Rayleigh-Sommerfeld propagation



# **Part I**

## **Introduction**



# Chapter 1

## Introduction

### 1.1 Preamble

With the recent development of immersive and collaborative environments, consumers seek a better immersion during their remote interactions. Telepresence systems already include three-dimensional (3D) audio effect and life-sized participants display. Therefore, the ability to provide 3D visualization with natural and realistic depth illusion has become one of the most anticipated features for these systems [145].

Most current 3DTV systems are based on the stereoscopic technique. This technique provides 3D illusion by delivering two different views of the scene to the respective left and right eyes of the viewer. The viewer thus perceives the 3D effect by the binocular parallax activity. Thanks to its simplicity of implementation and its good compatibility with existing 2DTV systems, the stereoscopic technique quickly attracted a considerable attention [193]. However, this technology still presents many constraints and limitations such as the necessity of wearing special glasses for stereoscopic systems and the limited viewing zone of auto-stereoscopic systems [32]. Moreover, the stereoscopic technique does not provide monocular motion parallax and creates a mismatch between convergence and accommodation cues, which can lead to eye-strain and headaches [56].

To solve these limitations, several alternative technologies have been proposed in the last decades. Among these techniques, holography is often considered as the most promising one, since it can produce the most realistic and natural 3D illusion. Indeed, it provides all the Human Visual System (HVS) depth cues without the need for special viewing devices and without causing eye-strain [157]. To create the depth illusion, a hologram diffracts an illuminating light beam to give it the shape of the light wave that would be transmitted or reflected by a given scene. As a consequence, the viewer perceives the scene as if it was physically present in front of him. However, a hologram is optically recorded by wave

interference between two coherent laser beams in a dark room. The optical system must be kept very stable during hologram recording, since a very small vibration can destroy the interference fringes. Because of these requirements, conventional optical holography cannot be easily used for video and outdoor recording.

To overcome optical hologram recording limitations, several methods have been proposed to generate holograms by computer calculation. Using these methods, it is possible to obtain Computer-Generated Holograms (CGH) of synthetic or existing scenes by simulating the propagation of light scattered by the scene towards the hologram plane. This work falls within this context and aims at addressing the challenges encountered by researchers in this area.

## 1.2 Objectives and motivations

Thanks to its attractive visualization properties, CGH may replace the stereoscopic technique in the future. To fulfill its promises, CGH computation techniques should meet several requirements. First, CGH computation methods should provide all the HVS depth cues to produce a strong depth illusion. Second, the displayed 3D scene should be as realistic as possible, which means that occlusions and shading should be accurately reproduced. Last but not least, to have application in the field of videoconferencing or telepresence systems, computation techniques should be able to produce CGH of real scenes at interactive rates.

However, since holograms create the depth illusion by diffracting an illuminating light beam, the viewing angle of a CGH is dependent on its pixel pitch, which should be on the order of the wavelength of visible light. Because of this microscopic size, a full-color CGH with large size and wide viewing angle requires a huge amount of information. For instance, a monochromatic red CGH (wavelength 640nm) of size (10cm  $\times$  10cm) with full-parallax and viewing angle of (120°  $\times$  120°) would require a pixel pitch of 369.5nm and a resolution of more than (270K  $\times$  270K). A full-color CGH of the same size combined by RGB channels would require at least three times more information.

Because of this huge amount of data to process, meeting all these requirements is still very challenging for state-of-the-art methods. As a consequence, most CGH computation methods make a trade-off between 3D scene realism, depth accuracy and calculation time. For example, to speed up the computation, some CGH do not provide all the human depth cues or use a simplified shading of the scene. The objective of this thesis is therefore to propose new algorithms able to produce CGHs of realistic scenes with strong depth illusion and reduced computational complexity.

## 1.3 Contributions

The contributions of this thesis are organized into two parts. The first part of this work aims at reducing the CGH calculation time to get closer to real-time computation. To this end, we designed and proposed two novel algorithms:

- **A fast hybrid point-source/wave-field approach:** This method enables efficient calculation of colorful hologram from synthetic 3D model or 2D-plus-depth data by combining two commonly used approaches: the point-source and wave-field approaches. Whereas state-of-the-art methods reduced the computational complexity of these approaches independently, this method uses them together and therefore takes advantages from both of them. The algorithm consists of three steps. First, the 3D scene is sliced into several depth layers parallel to the hologram plane. Then, for each layer, the complex wave scattered by this layer is computed either using a wave-field or a point-source approach according to a threshold criterion on the number of scene points within the layer. Finally, the complex waves scattered by all the depth layers are summed up to obtain the final CGH. A thorough analysis of the method reveals that this combination of approaches produces images without visible artifact and outperforms both the point-source and wave-field approaches.
- **A temporal redundancy removal algorithm:** This method computes colorful video holograms from 2D-plus-depth data at interactive rates by removing temporal redundancies between consecutive frames of the video. For each frame, the algorithm detects changes in the scene and updates the hologram of only affected scene points. Since only small regions of the hologram are updated at each video frame, this method allows the computational burden to be dramatically reduced, enabling the computation of colorful video holograms at 60 frames per second.

The second part of this work focuses on 3D scene realism. We proposed two algorithms to enhance the visual quality of displayed scenes:

- **A fast occlusion processing algorithm:** This method aims at improving the visual quality of the displayed scene by taking into account occlusions between objects during the calculation of a hologram from synthetic 3D models. To this end, we designed an efficient algorithm for light shielding between points and light waves, allowing the occlusion of wave-field layers by point-source layers and vice-versa. Experimental results revealed that occlusions are accurately reproduced using this method without significantly increasing the hologram calculation time of the original hybrid method.

- **A method for rendering specular reflections:** This method enables the calculation of colorful holograms with specular reflections from Multiview-plus-Depth (MVD) data. The algorithm consists of three steps. First, intensity views and associated depth maps of the scene are taken from different perspective viewpoints. Then, the 3D scene geometry is reconstructed from the MVD data. This 3D scene reconstruction step allows us to use only a few perspective projections of the scene without sacrificing any depth cue. Furthermore, to take into account specular reflections, each scene point is considered to emit light differently in all the directions. Finally, light scattered by the scene is numerically propagated towards the hologram plane to get the final CGH. Experimental results show that this method is able to provide all the human depth cues and accurate shading of the scene without visible artifact.

## 1.4 Thesis organization

Chapter 2 includes background information essential to the development of the thesis. First, we define the objectives and requirements of 3D video systems. Then, some theoretical background on conventional optical holography and the full processing chain of CGH computation methods are provided. Finally, we give a brief background on scalar diffraction theory, which is the backbone of CGH systems.

Chapter 3 provides a complete state-of-the-art of hologram synthesis methods. First, we define the hologram synthesis parameters and give a brief discussion on how to compare CGH computation methods. Then, we present various algorithms to compute the light wave scattered by the scene into the hologram plane. Finally, we give a brief review of methods to encode the resulting complex light field into a hologram with real and positive values.

Chapter 4 describes our first contribution, dedicated to the reduction of CGH calculation time using a hybrid point-source/wave-field approach. We first develop the proposed method. Then, a detailed description of the Graphics Processing Unit (GPU) implementation is provided. Finally, the experimental results with respect to various recording parameters are analyzed.

Chapter 5 develops our second contribution, which combines temporal redundancy removal and hybrid point-source/wave-field approaches to further reduce the calculation time of video holograms. We first present the proposed method. Then, we provide a thorough analysis of the CGH computation time with respect to various 3D video contents.

Chapter 6 describes our third contribution, which aims at improving the visual quality of the displayed scene by taking into account occlusions between scene objects. First, we

---

present the proposed method and provide a detailed description of the GPU implementation. Finally, the experimental results are analyzed.

Chapter 7 develops our last contribution, dedicated to the rendering of specular reflections. First, we present a preliminary version of the proposed method which can only render diffuse scenes. Then, we generalize this method to scenes with specular reflections. Finally, experimental results are analyzed.

Chapter 8 concludes this document. The initial targets are first reminded and the directions taken are discussed. Then, a summary of achieved targets is provided, and the future work prospects and potential improvements are addressed.

Finally, Appendix A lists the publications and patent published during the thesis.



## **Part II**

# **Background**



# Chapter 2

## Background

This chapter aims at providing necessary background information for understanding the thesis. The objectives and requirements of 3D video systems are first described in Section 2.1. Section 2.2 then provides a theoretical background on conventional optical holography and Section 2.3 describes the full processing chain of CGH systems. Finally, Section 2.4 gives a brief background on scalar diffraction theory, which is the backbone of CGH computation methods.

### 2.1 Objectives and requirements of 3D video systems

The goal of 3D video systems is to produce natural and realistic depth illusion to the human eye. To overcome the limitations of current 3DTV systems, we need to understand how the HVS perceives the 3D effect. This section first defines the HVS depth cues and their meaning. Then, the realization of these cues in 3D visualization systems is described.

#### 2.1.1 Human Visual System depth cues

The HVS relies on a large number of cues to estimate the distance and shape of objects located in 3D space [146]. As illustrated in Figure 2.1, these depth cues can be classified into two categories: the binocular and monocular depth cues [52].

The binocular depth cues require the activity of both eyes to be perceived. These include retinal disparity and vergence, which are described in the following.

**Retinal disparity:** Because of the horizontal separation of our eyes, each eye perceives the world from a slightly different perspective. This difference in the perceived images is called retinal disparity and is the most important depth cue for medium viewing distances.

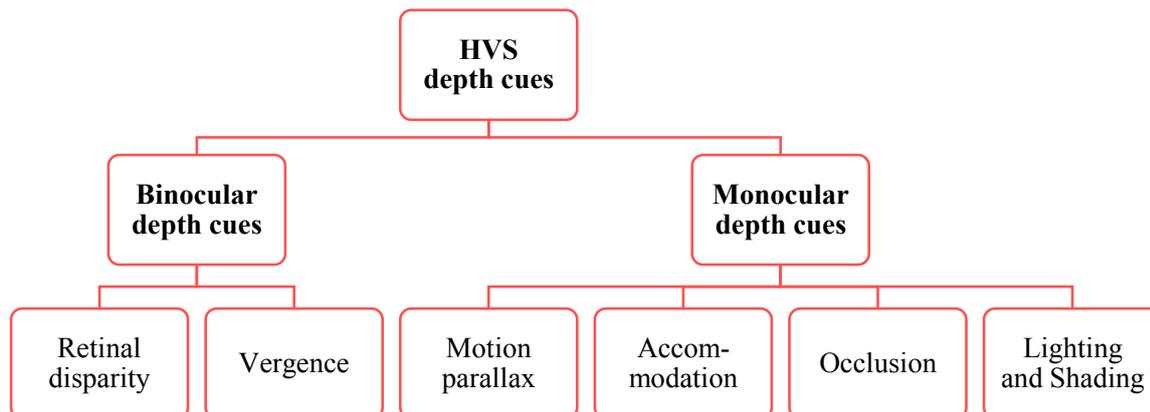


Fig. 2.1 Overview and classification of human visual system depth cues.

**Vergence:** The vergence corresponds to the angle between the optical axes of our eyes. This angle is small when watching at far objects and becomes more important when watching at closer objects. This oculomotor cue also helps our brain in depth perception but is only effective for short distances.

The monocular depth cues can be perceived with just one eye, and are at least as important in depth perception as binocular cues [27]. In addition to pictorial depth cues, which can be observed in 2D images, monocular depth cues also include motion parallax and accommodation. Their meaning is given in the following.

**Motion parallax:** When we shift our viewpoint sideways, nearby objects move faster in our field of vision than farther objects. These relative movements of objects correspond to motion parallax, which is an important HVS depth cue for medium viewing distances.

**Accommodation:** Accommodation is the tension of ciliary muscles that changes the focal length of our eyes to keep objects in sharp focus on the retina. This oculomotor cue is coupled with the vergence cue and is only effective for short distances.

**Occlusion:** Occlusion is the strongest pictorial depth cue. If one object partially blocks the view of another object, we perceive it as closer. This cue only gives information about relative depths of objects.

**Lighting and shading:** The shape and position of objects can also be determined by the way light is reflected by object surfaces and by the shadows they cast.

Monocular depth cues also include other pictorial cues, such as linear perspective, aerial perspective, relative and familiar size, texture gradient or elevation. These are thoroughly described in [131].

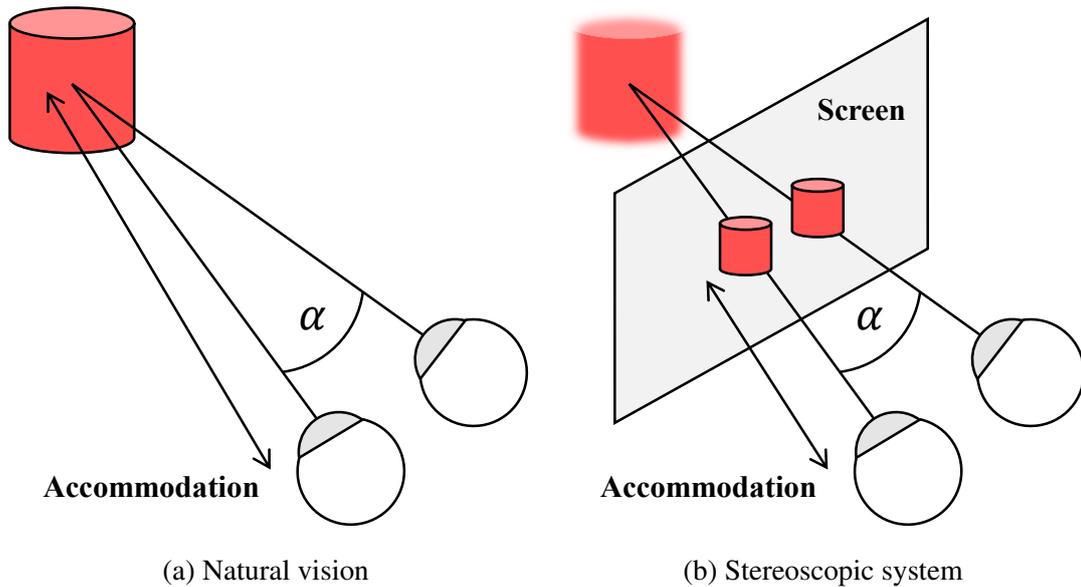


Fig. 2.2 Accommodation/vergence conflict in stereoscopic systems.

### 2.1.2 The stereoscopic technique

All the depth cues listed in the previous section are automatically perceived by the HVS when watching a real scene. As a consequence, to provide a natural and realistic depth illusion, 3D displays have to reproduce these depth cues to the end user correctly. As stated before, the pictorial depth cues such as occlusion or lighting and shading are also observed in 2D images. However, to provide retinal disparity, vergence, motion parallax and accommodation cues, 3DTV systems have to send more than one view of the scene to the eyes of the viewer.

Due to its simplicity of implementation and its good compatibility with existing 2DTV systems, the stereoscopic approach is the most widely used technology in current 3DTV systems. Stereoscopic displays create the depth illusion by delivering two different views of the scene to the respective left and right eyes of the viewer, thus providing both the retinal disparity and vergence cues, in addition to the pictorial depth cues. Motion parallax can also be simulated with the addition of a head-tracking system [30]. Thanks to the use of a lenticular array or a parallax barrier, auto-stereoscopic displays do not require any special viewing device. Moreover, by delivering more than two views of the scene, auto-stereoscopic displays can also provide the motion parallax cue without using any head-tracking system.

However, since each eye of the viewer only perceives a 2D projection view of the 3D scene, (auto-)stereoscopic displays cannot provide the accommodation cue. Indeed, to perceive sharp retinal images, the viewer always has to focus on the screen, where these images are displayed. This means that the depth information provided by the accommodation

cue corresponds to the distance of the screen, rather than the actual location of the 3D object. Since the accommodation cue gives an incorrect depth information, the perceived 3D effect is not as realistic as in natural vision. Moreover, as stated before, the accommodation and vergence cues are coupled with each other and always provide the same depth information in natural vision. However, in stereoscopic systems, the consistency between accommodation and vergence cues is broken, as shown in Figure 2.2. This mismatch between these two depth cues is commonly called the accommodation/vergence conflict and leads to visual discomfort, eyestrain, and headaches [56].

### 2.1.3 Alternative technologies

The accommodation cue is thus essential for providing natural depth illusion. Several 3D visualization technologies are able to provide this depth cue. These include integral imaging, volumetric and holographic displays.

Integral imaging systems provide the accommodation cue by delivering several views of the scene to each eye of the viewer using a lenticular array [101]. However, since each lens delivers a single view of the scene to the eyes of the viewer, the lenses have to be small enough to provide the accommodation cue, leading to a poor imaging resolution.

Volumetric displays provide the accommodation cue by illuminating scene points at their actual position in space. Two approaches are commonly used: the swept-volume and static-volume displays. In swept-volume displays, the 3D scene is decomposed into a series of 2D slices which are projected onto or from a display surface undergoing motion. Due to the persistence of vision, the viewer thus perceives a continuous 3D image. Static-volume displays create the depth illusion by filling the 3D space with addressable active elements called voxels, which are either transparent or luminous depending on their state. By changing the state of each voxel, it is thus possible to construct a real 3D image. However, volumetric displays can only create the 3D image in a limited volume and do not provide the occlusion cue, which is the strongest pictorial depth cue [27].

By contrast, holography is often considered as the most promising 3D display since it can provide every human depth cues in an unlimited 3D volume [157]. This is the only technology able to record the 3D appearance of the real world onto a 2D recording medium and to accurately reproduce it to our naked eyes. To create the depth illusion, a hologram diffracts an illuminating light beam to give it the shape of the light wave that would be transmitted or reflected by a given scene. As a consequence, the viewer perceives the scene as if it was physically present in front of him.

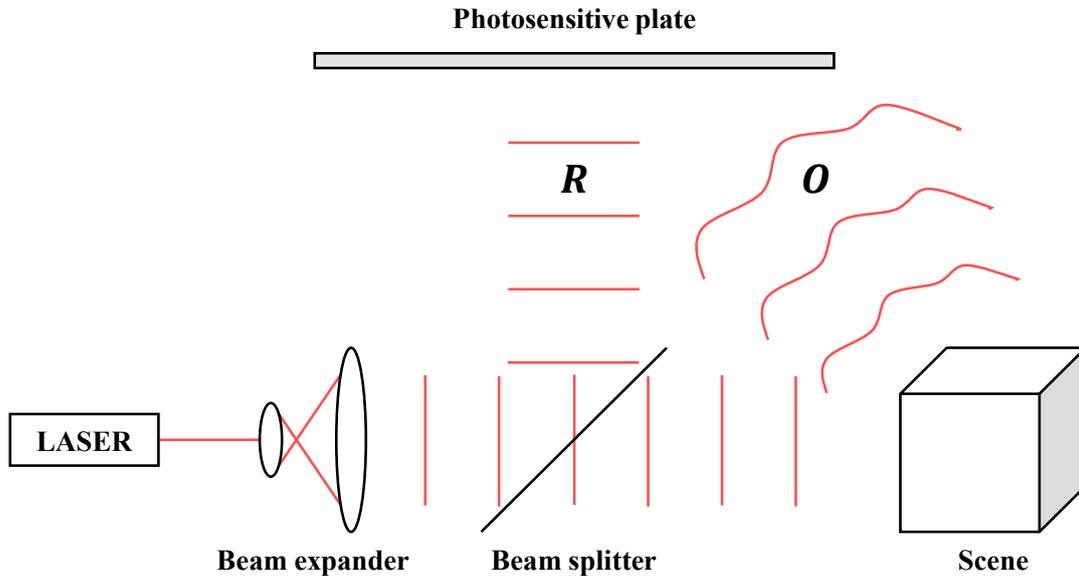


Fig. 2.3 Optical hologram recording set-up.

## 2.2 Conventional Optical Holography

Holography was invented in 1948 by Hungarian physicist Dennis Gabor while doing research in electron microscopy [42]. However, this technique only became practical with the advent of the first laser in the 1960's [97]. While conventional photography only enables to capture the energy of light, which is proportional to its intensity, a hologram records both the amplitude and phase of the light wave scattered by a scene enlighten by a laser beam. When the hologram is illuminated by the laser used during recording, the original light wave is reproduced, allowing the viewer to perceive the scene as if it was physically present in front of him. Holography thus provides a natural and realistic depth illusion.

In this section, we give a brief background on conventional optical holography. A thorough presentation of the principles of holography can be found in [140]. Conventional optical holography comprises a recording and a reconstruction steps.

### 2.2.1 Optical hologram acquisition

The general set-up of the optical acquisition process is shown in Figure 2.3. The coherent beam emitted by the laser is split into two waves: the first one, called object wave, is diffracted and/or reflected by the scene toward the photosensitive plate, while the second, called reference wave, illuminates the plate directly. These two waves produce interference

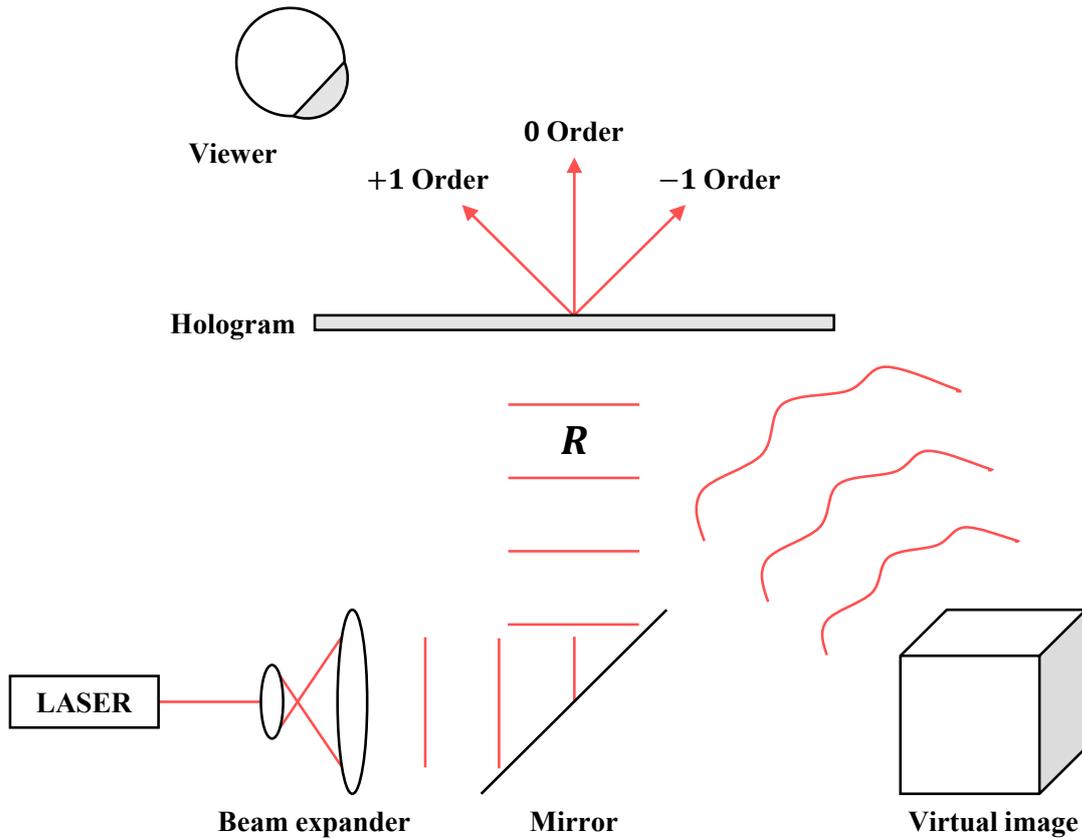


Fig. 2.4 Optical hologram reconstruction set-up.

fringes which are recorded by chemical development of the photosensitive medium. This interference pattern is called a hologram.

We call  $O = |O| \exp(j\phi(O))$  the object wave and  $R = |R| \exp(j\phi(R))$  the reference wave in the hologram plane.  $|C|$  and  $\phi(C)$  are the amplitude and phase of complex number  $C$ , respectively. The hologram recorded onto the photosensitive plate is therefore given by

$$\begin{aligned} H &= K(O + R)(O + R)^* \\ &= K|O|^2 + K|R|^2 + KOR^* + KRO^*, \end{aligned} \quad (2.1)$$

where  $K$  is a constant and  $C^*$  is the complex conjugate of  $C$ .

As shown in Eq. (2.1), both the amplitude and phase of the object wave have been encoded onto the hologram.

### 2.2.2 Optical hologram reconstruction

The general set-up of the optical reconstruction process is shown in Figure 2.4. The hologram is illuminated by the reference wave  $R$  used during the acquisition step. The light wave transmitted by the hologram is therefore given by

$$\begin{aligned}
 U &= H \times R \\
 &= \underbrace{K(|O|^2 + |R|^2)}_{0 \text{ diffraction order}} R + \underbrace{K|R|^2 O}_{+1 \text{ diffraction order}} + \underbrace{KR^2 O^*}_{-1 \text{ diffraction order}}. \quad (2.2)
 \end{aligned}$$

The light wave transmitted by the hologram includes three terms. The first term, called zero diffraction order, is proportional to the reference wave  $R$ . It corresponds to the undiffracted wave passing through the hologram. The second term, called "virtual image", or +1 diffraction order, is proportional to the object wave  $O$ . It produces an image of the original scene with all the effects of perspective and depth of focus. This term enables the viewer to see the scene as if it was physically present in front of him. The third term, called "real image", or -1 diffraction order, is proportional to the complex conjugate of the object wave  $O^*$ . It produces an image of the scene with an inverted depth.

The zero diffraction order and the real image are unwanted artifacts that may overlap with the reconstructed object wave. However, when the reference wave and the object wave illuminate the hologram with different incidence angles during the acquisition step, these three waves are transmitted at different angles during the reconstruction, thus enabling the 3D image of the object to be clearly observed [98].

### 2.2.3 Constraints and limitations

Despite its attractive 3D visualization properties, conventional optical holography still presents many constraints.

First, to properly record the interference fringes, the scene has to be exclusively illuminated by the laser source. As a consequence, to avoid interferences with external light, the hologram recording step is usually performed in a dark room. Moreover, since the slightest vibration can destroy the interference fringes, the optical system must remain extremely stable during the acquisition. Because of these requirements, conventional optical holography cannot be easily used for video and outdoor recording.

To avoid these limitations, several methods have been proposed in the last decades to replace the optical hologram acquisition process by numerical calculations. Using these methods, it is possible to generate a hologram of a synthetic or real scene without the need for the actual physical process of interference between light waves.

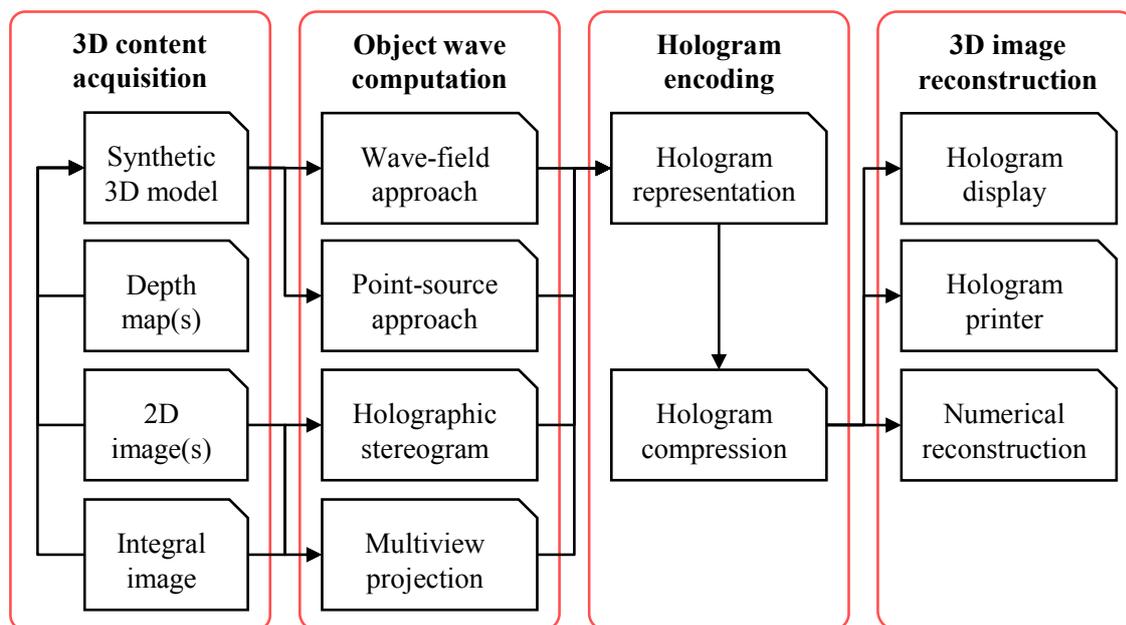


Fig. 2.5 Full Computer-Generated Holography processing chain.

## 2.3 Computer-Generated Holography

Computer-Generated Holography aims at replacing the optical acquisition process by numerical calculations. Figure 2.5 shows the full processing chain of CGH systems, which is composed of four blocks: 3D content acquisition, object wave computation, hologram encoding and 3D image reconstruction. The detailed description of these steps is given in the following.

### 2.3.1 3D content creation

The first step of CGH systems is to acquire the geometry of the input 3D scene. The easiest way to do this is to create a fully synthetic 3D scene via some specialized software. By this way, the 3D scene geometry, the objects materials and location of light sources can be accurately described. However, to compute holograms of real existing scenes, it is necessary to use a 3D capture setup.

Three approaches are commonly used to capture the 3D geometry of a real existing scene. The first one is to acquire 2D projections of the scene from different perspective viewpoints using a 2D camera array or by shifting a single camera along the vertical and horizontal axes. Another approach is to set up one or more 2D-plus-depth cameras to obtain the 2D views and associated per-pixel depth information. Finally, it is also possible to use integral imaging

systems, such as plenoptic cameras, to simultaneously capture a large number of projections in a single camera shot.

Regardless of the recording set-up, the acquired data can be directly used to generate the hologram of the scene. However, it is also possible to use 3D reconstruction algorithms to obtain the 3D model of the scene from the acquired data before any CGH calculation [122].

### 2.3.2 Object wave computation

The second step of CGH systems is to compute the object wave  $O$  scattered by the scene into the hologram plane. Depending on the type of data used to represent the 3D scene geometry, four main approaches are commonly used. These include wave-field approach, point-source approach, holographic stereograms, and multiple-viewpoint-projection approach. These methods are built upon scalar diffraction theory, which is briefly introduced in Section 2.4. State-of-the-art methods for object wave computation are thoroughly described in Chapter 3.

### 2.3.3 Hologram encoding

The third step of CGH systems is to adopt a representation of the object wave that can be encoded into a hologram. Indeed, the object wave  $O$  is a complex-valued field containing real and imaginary parts. However, to be displayed on a screen or printed onto a transparency, the hologram should have only real positive values. Moreover, these values need to be quantized to the number of levels given by the characteristics of the plotting device. Three representations are commonly used to encode the object wave into a real-valued hologram. These include amplitude holograms, which modulate the amplitude of the reference wave, phase holograms, which modulate its phase, and complex modulation holograms, which control both the amplitude and phase. State-of-the-art methods for hologram representation are described in Chapter 3.

Once a suitable hologram representation has been chosen, it is necessary to use data compression techniques to reduce the amount of holographic information to be stored and transmitted. Indeed, CGHs contain a massive amount of data: for instance, a monochromatic red CGH (wavelength 640nm) of size (10cm  $\times$  10cm) with full-parallax and viewing angle of (120°  $\times$  120°) contains more than 73GB of information. However, since holograms have very different signal properties compared to 2D images and videos, conventional compression techniques are not suited to holographic data, and specific compression algorithms have to be used. Several methods have been proposed in the last decade, including quantization-based methods [119, 169, 171, 198], transform-based methods [29, 170, 187, 188], and methods

based on standard coding schemes [4, 13, 28, 194, 195, 197]. An overview of existing holographic data compression techniques can be found in [37] and [196].

### 2.3.4 3D image reconstruction

Once the CGH has been computed and encoded, several techniques can be used to reproduce the 3D image of the original scene.

The most commonly used approach is to display the computed hologram on a beam shaping device called Spatial Light Modulator (SLM). SLMs are electrically or optically addressed displays which reproduce the original object wave by modulating the amplitude and/or the phase of the reference beam. Several types of SLMs can be used in holographic displays, including Liquid Crystal on Silicon (LCoS) [84], Digital Micro-mirror Displays (DMD) [178] and Acousto-Optics Modulators (AOM) [89]. Regardless of the technology they use, SLMs need to have small pixels to achieve their diffractive behavior and provide a large viewing angle. However, current SLMs still have limited resolutions and pixel pitches of over 10 times the wavelength of visible light. These limitations prevent SLMs from being used for large holograms with wide viewing angle.

For example, the holographic display of the National Institute of Information and Communications Technology (NICT, Japan) uses liquid crystal SLMs of resolution ( $7680 \times 4320$ ) with a pixel pitch of  $4.8\mu\text{m}$ , which are able to display holograms of size ( $3.7\text{cm} \times 2.1\text{cm}$ ) with a viewing angle of  $5.6^\circ$  [208]. To increase the size of the hologram to be displayed, it is possible to physically tile multiple SLMs [108, 109]. However, building a system of seamless tiled SLMs is still very challenging, and the resolution of such systems is still insufficient for large holograms. An overview of existing holographic display systems can be found in [209].

In order to plot large CGHs with pixel pitch in the order of the wavelength of visible light, it is possible to print them onto a photosensitive plate using a wavefront printer [87, 126, 215]. Figure 2.6 shows the general set-up of a wavefront printer, which consists of four elements: a laser, whose beam passes through a shutter, a SLM, a demagnifying system composed of lenses L1 and L2, and finally the photosensitive plate, mounted on a X-Y stage. First, a segment of the CGH is displayed on the SLM. Then, the control system opens the shutter, and the SLM is illuminated by the laser beam, thus reproducing the object wave. The reconstructed object wave is demagnified by lenses L1 and L2 and exposed to the photosensitive plate, while the zero diffraction order and real image are removed by the band-pass filter. In addition, the photosensitive plate is directly illuminated by the laser beam from the other side. After the exposure, the shutter is closed and the plate is translated by the X-Y stage to the next exposure position, while the next segment of the CGH is displayed on the SLM. This process is repeated until the entire hologram is exposed.

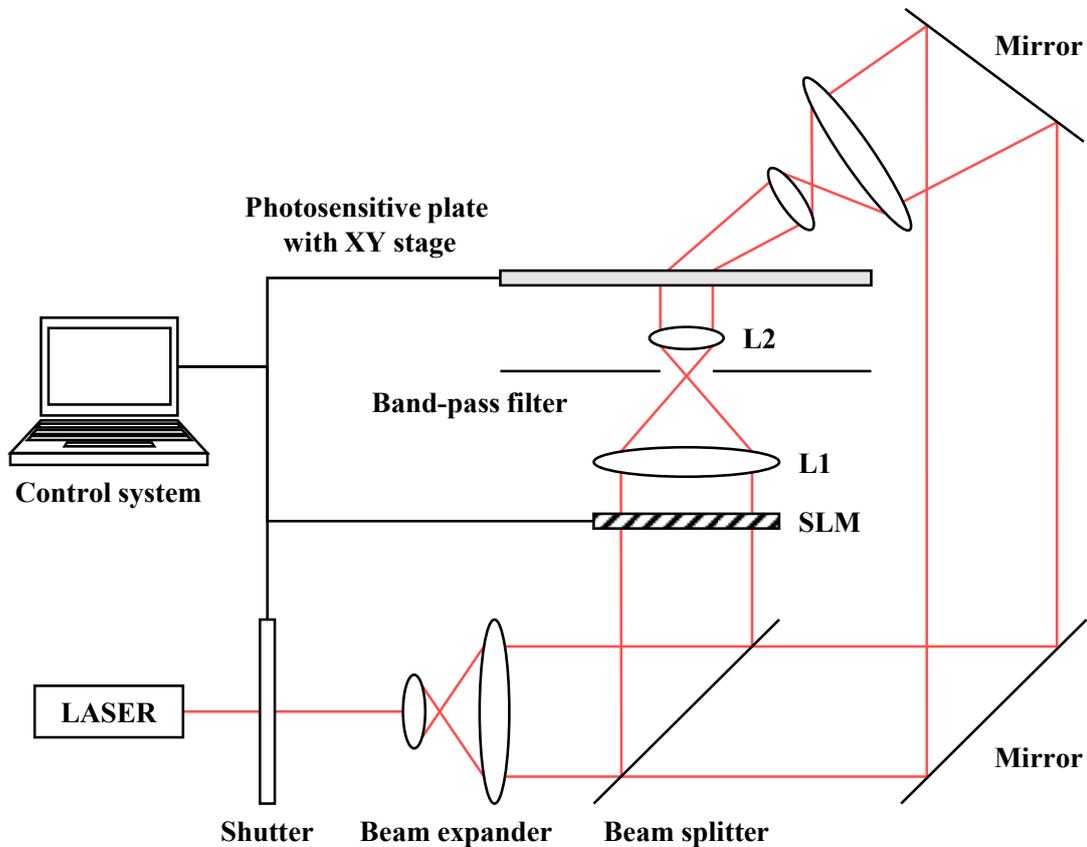


Fig. 2.6 Wavefront printer set-up.

Wavefront printers enable large CGHs with wide viewing angle to be printed. However, building such a system is not trivial and requires expensive material. As a consequence, most research labs do not have access to this kind of printer. Fortunately, to numerically verify the validity of large CGHs, it is possible to simulate the propagation of light wave transmitted by the hologram using diffraction formulas: this is called numerical reconstruction of holograms [141, 147, 186].

## 2.4 Scalar diffraction theory

In this section, we give a brief background on scalar diffraction theory, which is the backbone of CGH systems. Indeed, as discussed above, both the computation and numerical reconstruction of a hologram require the propagation of light to be simulated.

### 2.4.1 The Helmholtz equation

Diffraction is the behavior of light waves when they encounter an obstacle or a slit, which can be interpreted as the diffusion of the wave by the object points. It is thus closely related to light propagation. Diffraction was first described by Grimaldi in 1665 and interpreted as a wave phenomenon by Huygens in 1678. However, the first mathematical formulation of the wave theory of light was given by Maxwell in 1860, who identified light as an electromagnetic field.

In the following, we consider that the light wave is propagating in a dielectric medium that is linear, isotropic, homogeneous and nondispersive. The scalar diffraction theory treats the propagation of light in such a medium as a scalar phenomenon, thus neglecting the vectorial nature of electromagnetic fields [53]. This approximation yields the scalar wave equation, which is very accurate if both the diffracting aperture, i.e. the hologram in our case, and the propagation distance are large compared to the wavelength of light [172]. Fortunately, these conditions are well satisfied in the context of CGH systems.

If we consider light as a stationary wave, the scalar wave equation can be simplified to the time-independent partial differential equation

$$(\nabla^2 + k^2) U = 0, \quad (2.3)$$

which is known as the Helmholtz equation. In Eq. (2.3),  $U$  is the light wave,  $\nabla^2$  is the Laplace operator, and  $k$  is the wave number, given by

$$k = \frac{2\pi}{\lambda}, \quad (2.4)$$

where  $\lambda$  is the wavelength of light.

### 2.4.2 Rayleigh-Sommerfeld propagation

The first rigorous solution of the Helmholtz equation was given in 1896 by Sommerfeld [175], who expressed the propagation of light between two parallel planes separated by distance  $z$  as a convolution product, commonly known as the Rayleigh-Sommerfeld diffraction formula:

$$U(x, y) = \mathcal{P}_z^{RS}\{u\}(x, y) = \iint_{-\infty}^{+\infty} u(\xi, \eta) g(x - \xi, y - \eta, z) d\xi d\eta, \quad (2.5)$$

where  $(x, y)$  and  $(\xi, \eta)$  are the coordinates in the destination and source planes, respectively.

In Eq. (2.5),  $u$  is the complex light wave in the source plane and  $g$  is the convolution kernel, given by

$$g(x, y, z) = \frac{z \exp(jkr)}{2\pi r^2} \left( \frac{1}{r} - jk \right), \quad (2.6)$$

where  $j$  is the imaginary unit and  $r$  is given by

$$r = \sqrt{x^2 + y^2 + z^2}. \quad (2.7)$$

### 2.4.3 Angular Spectrum propagation

According to the convolution theorem, Eq. (2.5) can be formulated as

$$U(x, y) = \mathcal{P}_z^{AS}\{u\}(x, y) = \mathcal{F}^{-1}\{\mathcal{F}\{u(\xi, \eta)\}G(f_x, f_y, z)\}, \quad (2.8)$$

where  $\mathcal{F}$  is the forward Fourier transform,  $f_x$  and  $f_y$  are the spatial frequencies, and  $G$  is the transfer function of the medium, given by

$$G(f_x, f_y, z) = \mathcal{F}\{g(x, y, z)\} = \exp\left(j2\pi z \sqrt{\lambda^{-2} - f_x^2 - f_y^2}\right). \quad (2.9)$$

This formulation of the propagation phenomena as a linear and invariant system is commonly known as the Angular Spectrum propagation. It must be noted that the Rayleigh-Sommerfeld and Angular Spectrum propagation formulas give exact results for both near and far fields without any approximation. In order to simplify these equations to less computationally expensive formulations, it is possible to introduce several approximations for the convolution kernel  $g$ .

### 2.4.4 The Fresnel approximation

In the Fresnel approximation, the distance between the source and destination planes is assumed to be large compared to the wavelength of light, i.e.  $z \gg \lambda$ . In that case, we can take

$$\frac{1}{r} - jk \approx -jk. \quad (2.10)$$

Furthermore, the distance  $r$  given by Eq. (2.7) can be expanded to a Taylor series, such that

$$r = z + \frac{x^2 + y^2}{2z} + \frac{[(x^2 + y^2)]^2}{8z^3} + \dots \quad (2.11)$$

In Eq. (2.6), the factor  $r^2$  appearing in the denominator is approximated with only the zeroth-order term of Eq. (2.11), while the factor  $r$  in the phase  $\exp(jkr)$  is approximated with the

zero and first order terms. The convolution kernel is therefore simplified as

$$g'(x, y, z) = \frac{e^{jkz}}{j\lambda z} \exp\left(\frac{jk}{2z}(x^2 + y^2)\right). \quad (2.12)$$

Substituting Eq. (2.12) into Eq. (2.5) yields the convolution form of the Fresnel approximation, given by

$$\begin{aligned} U(x, y) &= \mathcal{P}_z^{Fr1}\{u\}(x, y) = \iint_{-\infty}^{+\infty} u(\xi, \eta) g'(x - \xi, y - \eta, z) d\xi d\eta \\ &= \frac{e^{jkz}}{j\lambda z} \iint_{-\infty}^{+\infty} u(\xi, \eta) \exp\left(\frac{jk}{2z}((x - \xi)^2 + (y - \eta)^2)\right) d\xi d\eta. \end{aligned} \quad (2.13)$$

Another formulation of the Fresnel approximation is found by polynomial expansion of the products inside the exponential term in Eq. (2.13), yielding

$$\begin{aligned} U(x, y) &= \mathcal{P}_z^{Fr2}\{u\}(x, y) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}(x^2+y^2)} \iint_{-\infty}^{+\infty} u(\xi, \eta) e^{j\frac{k}{2z}(\xi^2+\eta^2)} e^{-j\frac{2\pi}{\lambda z}(x\xi+y\eta)} d\xi d\eta \\ &= \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}(x^2+y^2)} \mathcal{F}\left\{u(\xi, \eta) e^{j\frac{k}{2z}(\xi^2+\eta^2)}\right\}, \end{aligned} \quad (2.14)$$

which is seen to be proportional to the Fourier transform of the field in the source plane multiplied by a quadratic phase factor.

## 2.4.5 The Fraunhofer approximation

In addition to the Fresnel approximation, it is possible to further simplify the light propagation calculation by using the stronger Fraunhofer approximation. If the propagation distance  $z$  is large enough to satisfy the condition

$$z \gg \frac{k(\xi^2 + \eta^2)_{\max}}{2}, \quad (2.15)$$

then the quadratic phase factor in Eq. (2.14) is approximated to be equal to unity, yielding the Fraunhofer approximation

$$U(x, y) = \mathcal{P}_z^{Fh}\{u\}(x, y) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}(x^2+y^2)} \mathcal{F}\{u(\xi, \eta)\}, \quad (2.16)$$

which is proportional to the Fourier transform of the field in the source plane.

## 2.5 Conclusion

In this chapter, we first defined the HVS depth cues and their realization in current 3D visualization systems. Then, we presented the principles of conventional optical holography, including optical hologram acquisition and reconstruction set-ups, and we described the full processing chain of CGH computation methods. Finally, we provided a brief background on scalar diffraction theory, which is the backbone of CGH computation methods. All this background information is essential for understanding the following of the thesis.



# Chapter 3

## State of the art of hologram synthesis methods

This chapter provides a complete state-of-the-art of CGH computation methods. First, the hologram synthesis parameters and a discussion on how to compare CGH computation methods are given in Section 3.1. Section 3.2 presents various algorithms to compute the object wave scattered by the scene in the hologram plane, and Section 3.3 shows how to encode the resulting complex light field into a hologram with real and positive values.

### 3.1 Hologram synthesis parameters

The hologram calculation parameters can be classified into three categories: the parameters of the scene, the rendering parameters and finally the hologram parameters.

#### 3.1.1 Scene parameters

The scene parameters depend on the input data used and on the choices made for each method. These are described in the following.

**Scene representation:** As stated in Section 2.3, various types of input data can be used to generate a hologram: 3D models, 2D view projections, depth maps, etc. Depending on the chosen scene representation, the technical challenges are slightly different: while the 3D scene geometry can be accurately described when using synthetic 3D models, it may be necessary to use specific noise-reduction algorithms when using depth maps of real existing scenes.

**Scene resolution:** In points, polygons or 2D projection pixels, the resolution of the scene depends on the input data used: a real existing scene captured using an array of cameras often has a lower resolution than a purely synthetic scene. A high resolution allows for a great level of details in the scene but increases the computation time of the hologram.

**Scene size:** Displaying a large scene (over ten centimeters) is one of the biggest challenges of CGH systems. Indeed, the maximum size of the scene is limited by the dimensions of the hologram. Since the size of hologram pixels should be in the order of the wavelength of visible light, computing a large hologram requires a huge number of pixels and therefore a massive amount of memory. For this reason, many methods presented in this chapter have only been tested with input scenes of only a few centimeters or less.

### 3.1.2 Rendering parameters

The rendering parameters have a strong influence on the calculation time of the hologram. These are described in the following.

**Number of colors:** Depending on the number of lasers used for the restitution, a hologram can produce either a monochromatic or full-color scene. However, computing a full-color hologram is more time-consuming than computing a monochromatic hologram.

**Motion parallax:** Holography is able to provide motion parallax along the horizontal and vertical axes. Nevertheless, some methods restrict the parallax in the horizontal direction only, significantly reducing the computation time of the hologram at the cost of a less natural depth illusion.

**Occlusion processing:** Occlusion is one of the most important monocular depth cues. Therefore, the ability to properly take into account occlusions between objects in the scene is one of the key features of CGH computation methods. However, unlike still images, holograms provide motion parallax, which is coupled with the occlusion effect. Therefore, occlusion processing must be performed for all possible viewpoints, significantly increasing the CGH computation time. As a consequence, some methods described in this chapter use a very narrow field of view (and thus a reduced number of viewpoints) or perform occlusion processing for a single point of view. Moreover, some methods do not take into account occlusions between scene objects at all.

**Shading and lighting:** The shape and position of objects can be determined by the way light is reflected by object surfaces and by the shadows they cast. As a consequence, shading and lighting should be taken into account when computing a CGH of a given

scene. However, depending on the type of input data used, this process may require a huge computation time. For this reason, some methods do not perform any shading at all or use a simplified lighting of the scene.

**Computing system architecture:** All the CGH synthesis methods described in this chapter have been implemented and tested for specific computing systems with different architectures. As a consequence, an efficient algorithm for a given computing system may not be the most efficient for another system. For example, an algorithm with reduced computational complexity can perform very well when implemented on a single-core processor, but may be highly inefficient when implemented on a GPU if it is not parallelizable.

### 3.1.3 Hologram parameters

The hologram parameters depend on the characteristics of the optical system used for the restitution. These are described in the following.

**Hologram resolution:** The hologram resolution corresponds to the resolution of the optical system used for the restitution. As stated before, this parameter determines the maximum size of the scene and has a huge influence on the computation time and on the required memory.

**Pixel pitch:** The pixel pitch of the hologram corresponds to the pixel size of the optical system used for the restitution and determines the maximum field of view provided by the hologram. A smaller pixel pitch allows to provide a wider field of view but often increases the CGH computation time.

### 3.1.4 Comparison of hologram synthesis methods

Although all the methods presented in this chapter have the common goal of generating a hologram from a given scene, it is difficult to compare them. Indeed, to propose a classification of these methods, they should be compared with exactly the same calculation parameters and same input scenes. However, since no standardization has been performed in the field of Computer Generated Holography, each method has been designed to be used in a very specific context. As a consequence, comparing them with the same calculation parameters defined arbitrarily would be irrelevant. For this reason, in this chapter, we only compare the theoretical complexities of these methods.

Hologram synthesis methods generally consist of two steps: (1) calculation of the object wave, and (2) encoding of this complex wave into a hologram.

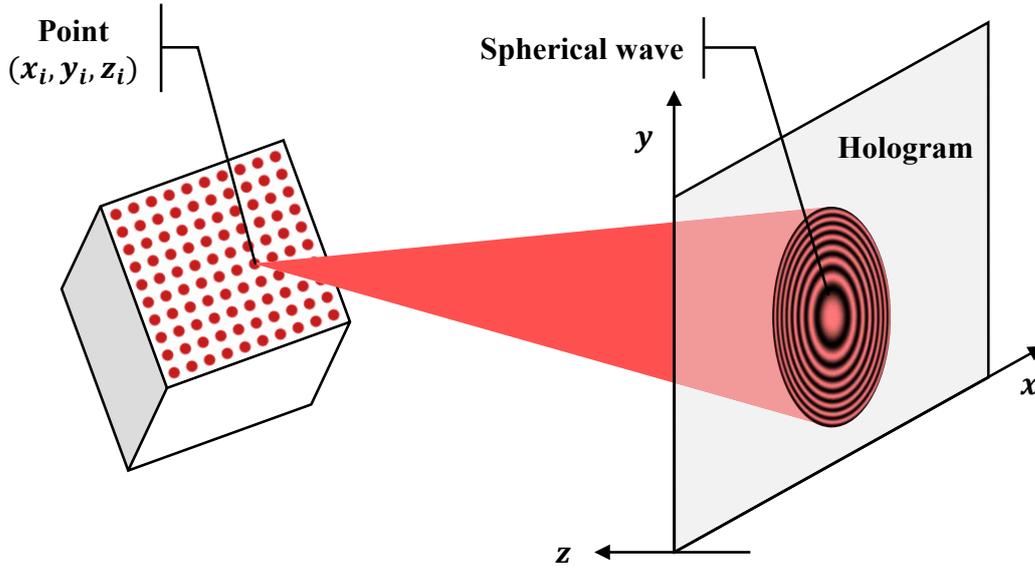


Fig. 3.1 Computation of the object wave using the point-source approach.

## 3.2 Object wave computation

The first step of CGH synthesis methods is to compute the object wave  $O$  scattered by the scene into the hologram plane. To this end, it is possible to use either a synthetic 3D model or multiple 2D view projections of the scene supplemented or not with depth maps.

### 3.2.1 Point-source approach

#### Original method

The point-source approach [15] is one of the most commonly used technique for object wave computation. Figure 3.1 shows the scene geometry and coordinate system used in this approach. The coordinate system is defined by  $(x, y, z)$  so that the hologram lies on the  $(x, y, 0)$  plane. The 3D scene is analytically described as a collection of isolated points, where each point is considered to be a spherical light source. The object wave scattered by the scene into the hologram plane can thus be computed as the sum of spherical waves scattered by each scene point, such that

$$O(x, y) = \sum_{i=0}^{M-1} \frac{A_i}{r_i} \exp[j(kr_i + \phi_i)] h(x - x_i, y - y_i, z_i), \quad (3.1)$$

where  $M$  is the total number of scene points,  $(x_i, y_i, z_i)$  are the coordinates of point  $i$ ,  $A_i$  is its amplitude, calculated using shading formulas, and  $\phi_i \in [0; 2\pi]$  is its initial phase, defined

randomly to avoid self-interference. Finally,  $r_i$  is the oblique distance, given by

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + z_i^2}. \quad (3.2)$$

In Eq. (3.1),  $h$  is a window function used to restrict the region of contribution of the point in the hologram plane, equal to 1 within the region of contribution and 0 elsewhere. This function limits the spatial frequencies of the complex wave to avoid aliasing in the CGH. According to the Nyquist Sampling Theorem, the maximum spatial frequency  $f_{\max}$  that can be represented with a sampling pitch  $p$  is given by  $f_{\max} = (2p)^{-1}$ . The grating equation [53] gives the relation between the maximum spatial frequency  $f_{\max}$  and the maximum diffraction angle  $\theta_{\max}$  as  $\sin(\theta_{\max}) = \lambda f_{\max}$ . Therefore, the region of contribution of a point at depth  $z$  is given by its maximum radius  $R_{\max}(z)$  such that

$$R_{\max}(z) = |z| \tan(\theta_{\max}) = |z| \tan\left(\arcsin\left(\frac{\lambda}{2p}\right)\right), \quad (3.3)$$

as shown in Figure 3.2.

The window function  $h$  can thus be defined as

$$h(x, y, z) = \begin{cases} 1 & \text{if } \sqrt{x^2 + y^2} < R_{\max}(z) \\ 0 & \text{otherwise.} \end{cases} \quad (3.4)$$

### Occlusion processing and shading

The point source approach is simple in principle and potentially the most flexible to generate the hologram of a 3D scene. Moreover, Computer Graphics (CG) shading techniques, such as Phong shading [139], can be integrated during the object wave computation to produce photo-realistic effects [91, 223], as shown in Figure 3.3. However, the original method

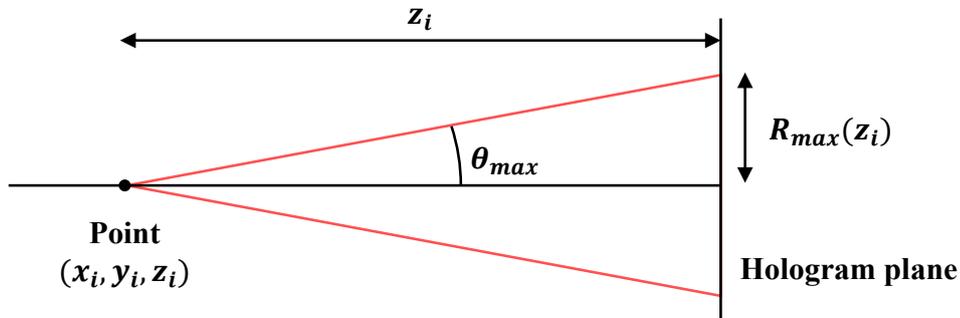


Fig. 3.2 Region of contribution of a given scene point.

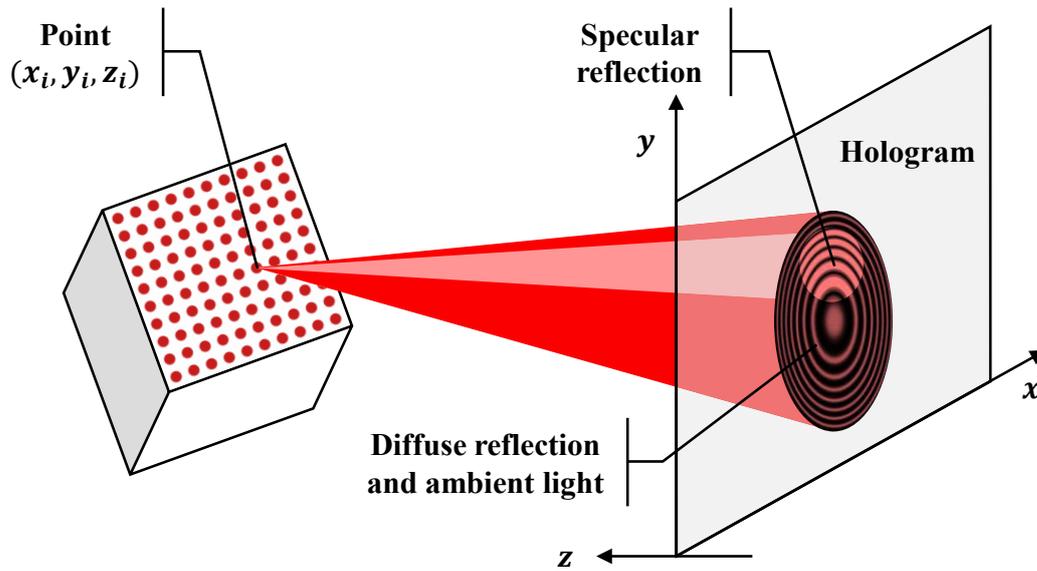


Fig. 3.3 Phong shading using the point-source approach.

proposed in [15] does not take into account occlusions between objects in the scene, reducing the realism of the rendered scene. Indeed, occlusion processing is necessary so that nearer surfaces and objects properly occlude those more distant. Two approaches are commonly used for occlusion processing in point-source based methods: the object- and image-oriented approaches.

In the object-oriented approach [185], the scene is first sampled as a collection of spherical light sources. Then, occlusion processing is performed as a visibility test to check the existence of obstacles between an object point and each pixel of the hologram, as shown in Figure 3.4. The computation time of this approach is thus dependent on the number of object points, and may be highly inefficient if many object points are not visible by any of the hologram pixels. To reduce the number of visibility tests, it is possible to divide the hologram pixels into groups that share the same set of visible scene points [20].

In the image-oriented approach [21, 221, 222], occlusion processing is performed using the ray casting technique, as shown in Figure 3.5. Evenly distributed rays are cast from each hologram pixel into the scene volume. The intersection of these rays with the scene produces a collection of points that are considered as spherical light sources. The contribution of these points to the corresponding hologram pixel is then calculated. The computation time of this approach is thus dependent on the number of rays for each hologram pixel. This approach works well when the 3D scene is located near to the hologram plane. However, since each bundle of rays is distributed in a cone, farther objects may be under-sampled, resulting in a

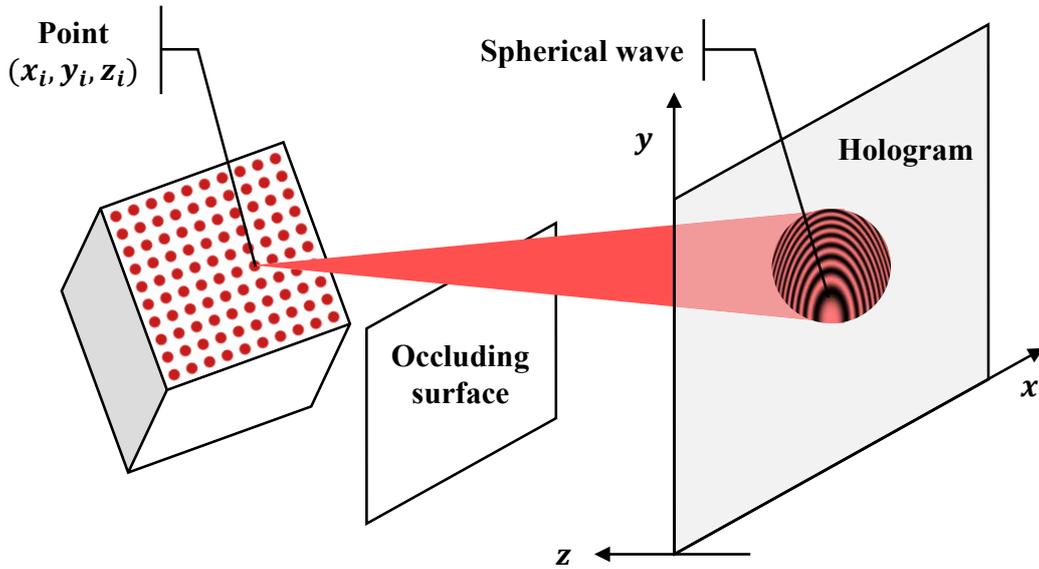


Fig. 3.4 Occlusion processing using an object-oriented approach.

poor imaging resolution. To reduce the number of rays, it is possible to divide the hologram pixels into groups that share the same bundle of rays [62].

The improved Hidden Surface Removal Method (HSRM) [41, 206] is another approach for occlusion processing which does not require any visibility test at all. In this approach, the hologram is divided into several rectangular patches. Then, perspective projection images and associated depth maps of the scene are captured from different viewpoints. These views and depth maps are then used to generate the collection of visible point sources for each sub-hologram. A technique which uses orthographic projection images has also been proposed in [67].

### Computation time reduction

Another limitation of the original point source method proposed in [15] is its high computational complexity since it requires one calculation per scene point per hologram pixel. Its computational complexity is therefore in the order of  $\alpha MN$ , where  $\alpha$  is the number of arithmetic operations,  $M$  is the number of scene points and  $N = (N_x \times N_y)$  is the number of hologram pixels. To reduce the computation load, three approaches can be used: (1) reduce the number of arithmetic operations  $\alpha$ , (2) reduce the number of pixels  $N$ , and (3) reduce the number of scene points  $M$  for which the spherical wave has to be computed.

The first approach to accelerate the object wave computation is to reduce the number of arithmetic operations  $\alpha$ . The calculation of the oblique distance given by Eq. (3.2) requiring

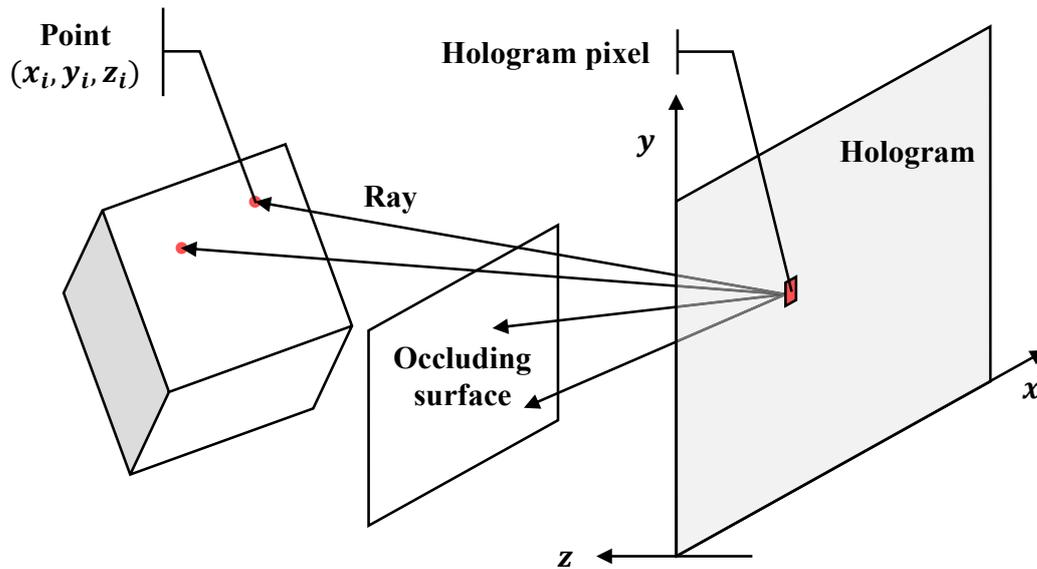


Fig. 3.5 Occlusion processing using an image-oriented approach.

a significant number of operations, several methods have been proposed to accelerate its computation. In [118] and [213], the authors propose to compute it using recurrence formulas between adjacent pixels in the hologram plane rather than time-consuming direct calculation at each hologram pixel. In [216], the authors propose to pre-compute this distance and to store it into a Look-up Table (LUT) indexed by  $(x, y, z)$ . This LUT is then simply addressed when computing the object wave. In [70], the authors propose to avoid redundant calculations by taking advantage of the geometric symmetry of the spherical light wave equation. In this method, the complex wave scattered by each point is simultaneously calculated at four different pixels in the hologram plane, dividing the object wave calculation time by 4.

Another way to reduce the number of arithmetic operations is to use a pre-computed LUT to store the spherical light wave scattered by scene points from each of the possible locations in the scene volume [107]. The entire object wave of a specific scene can thus be generated by fetching the corresponding waves to each object point from the LUT and adding them together. Although this method considerably accelerates the object wave computation, its greatest drawback is the enormous memory size of the LUT. In [81], the authors propose a method to reduce the memory usage of the LUT. In this method, the 3D scene is sliced into a set of depth layers parallel to the hologram plane, and only the waves scattered by the center-located scene points on each layer are pre-calculated and stored into the LUT. Then, the waves scattered by other scene points on each layer are generated using shifting and scaling operations. To further reduce its memory usage, several authors decompose the 3D LUT into smaller 2D or 1D LUTs using different trigonometric relations [44, 68, 83, 132].

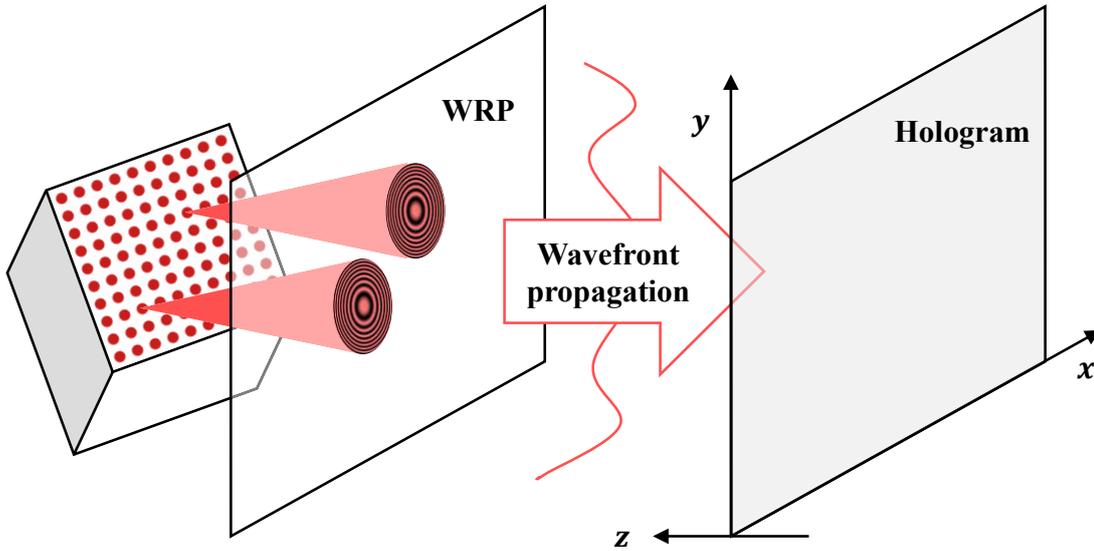


Fig. 3.6 Wavefront recording plane technique.

Alternatively, it is possible to take advantage of the geometric symmetry of the spherical light wave to store only a part of it into the LUT [24, 127].

The second approach to accelerate the object wave computation is to reduce the number of pixels  $N$  for which the spherical wave has to be computed. A straightforward way to do this is to locate the 3D scene very close to the hologram plane [205]. By this way, the region of contribution of each spherical light source in the hologram plane is small, and the number of calculations per scene point is reduced. Although this method considerably accelerates the object wave computation, for some applications it is not possible to locate the scene sufficiently close to the hologram plane.

Alternatively, it is possible to sample the object wave in an intermediate plane called Wavefront Recording Plane (WRP) and located close to the scene [165, 167, 192], as shown in Figure 3.6. The method consists of two steps. In the first step, the object wave scattered by the scene is recorded onto the WRP using Eq. (3.1). Since the WRP is placed close to the scene, the object light traverses small regions on the WRP, reducing the computational burden. In the second step, the object wave on the WRP is numerically propagated towards the hologram plane using one of the propagation formulas presented in Section 2.4, involving one to three Fourier transforms and a complex multiplication. In [130, 138, 168, 176], the authors employ multiple WRPs that are crossing through the scene to further reduce the distance of each scene point to the closest WRP.

The third approach to accelerate the object wave computation is to reduce the number of scene points  $M$  for which the spherical wave has to be computed. To this end, it is possible

to reduce the spatial redundancies between object points. Two methods have been proposed to do this. The first one uses Run-Length Encoding (RLE) algorithm to group adjacent scene points with same depth  $z_i$  and same intensity [82, 123]. The light waves scattered by scene points within a single group are then calculated all at once using a different LUT depending on the number of scene points. The second method uses Differential Pulse Code Modulation (DPCM) to reduce interline redundancy between the pixels of a 3D image [79]. The light wave scattered by each horizontal or vertical line of the image is computed by updating the light wave of the previous line depending on the differences between them. Finally, the waves scattered by each scene line are summed up together to obtain the complete object wave.

Another method to reduce the number of scene points  $N$  is to take advantage of temporal redundancies between consecutive frames of a 3D video. Indeed, there exist slight changes in intensity and depth between consecutive frames, and there is no need to recalculate the object wave for each frame. The principle is to detect changes in the scene and update the light waves of only affected scene points. Several methods have been proposed to track scene changes, including per-point incremental update [85, 142, 177], object-based motion compensation [33, 80], block-based motion compensation [34, 35] and object tracking mask-based method [94].

Finally, to further accelerate the object wave computation, it is possible to use GPU hardware [5, 93, 111, 132] and special purpose hardware [59, 64, 71].

### 3.2.2 Wave-field approach

#### Planar layers method

The wave-field approach based on a layered model of the scene [10, 105, 166, 227, 228] is another well-known technique for object wave computation. Figure 3.7 shows the scene geometry and coordinate system used in this approach. The coordinate system is defined by  $(x, y, z)$  so that the hologram lies on the  $(x, y, 0)$  plane. The 3D scene is sliced into a set of  $N_z$  depth layers parallel to the hologram plane and located between  $z_{\min}$  and  $z_{\max}$ . We call  $d_z$  the distance between each depth layer. The depth layers are numbered from 0 to  $N_z - 1$ , from the farthest to the nearest to the hologram plane.

Each depth layer  $d$  operates as a surface source of light which emits a complex wave  $o_d$  given by

$$o_d(x, y) = A_d(x, y) \exp [j\phi_d(x, y)], \quad (3.5)$$

where  $A_d(x, y)$  is the amplitude of the  $(x, y)$  point within layer  $d$ , calculated using illumination formulas, and  $\phi_d(x, y) \in [0; 2\pi]$  is its phase, set to a uniform random value to render a diffuse scene.

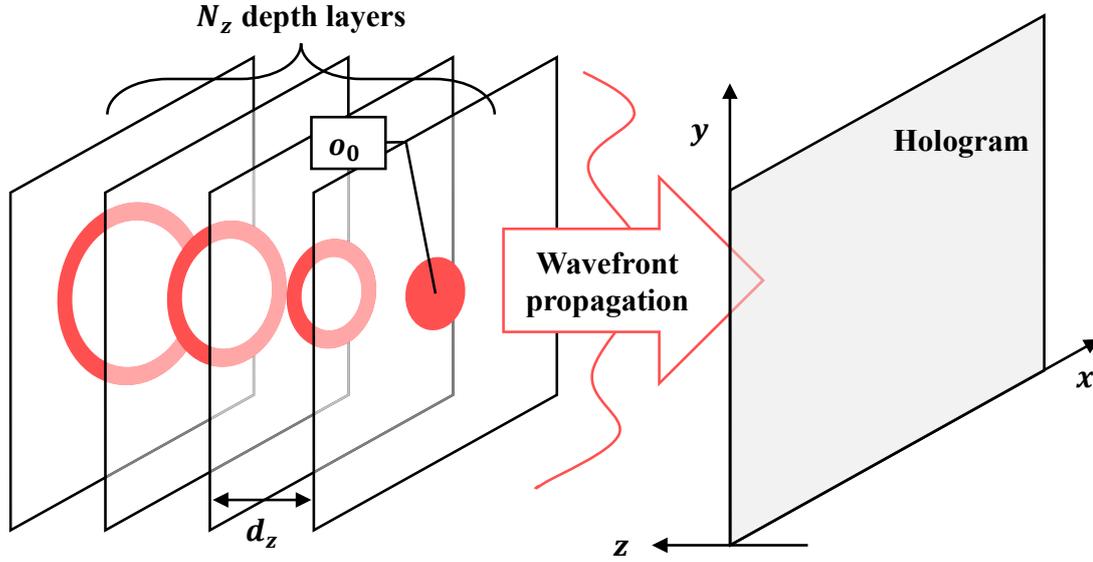


Fig. 3.7 Wave-field approach based on a layered model of the scene.

Light scattered by the scene is numerically propagated and shielded from one layer to another using the recurrence formula

$$\begin{cases} u_0(x, y) = o_0(x, y) \\ u_d(x, y) = o_d(x, y) + m_d(x, y)\mathcal{P}_{d_z}\{u_{d-1}\}(x, y) \quad \text{for } d > 0, \end{cases} \quad (3.6)$$

where  $u_d$  is the total complex wave scattered by layer  $d$  and operator  $\mathcal{P}_z$  stands for the numerical propagation of light between two parallel planes separated by a distance  $z$ , which can be computed using one of the diffraction formulas presented in Section 2.4. In Eq. (3.6), occlusion processing is performed using the binary cross-section mask function  $m_d$ , which has value 0 on the obstacles and 1 elsewhere. Occlusion processing is thus performed in a similar way as the painter's algorithm in CG.

Finally, the complex wave scattered by layer  $N_z - 1$  is numerically propagated to the hologram plane to obtain the object wave, such that

$$O(x, y) = \mathcal{P}_{z_{\min}}\{u_{N_z-1}\}(x, y). \quad (3.7)$$

### Generalization to polygon meshes

The planar layers method proposed in [105] is relatively simple to implement and yields accurate results for most 3D scenes. However, this method is not well suited when the depth extent of the scene is large. Indeed, to remain invisible to the human eye, the distance  $d_z$

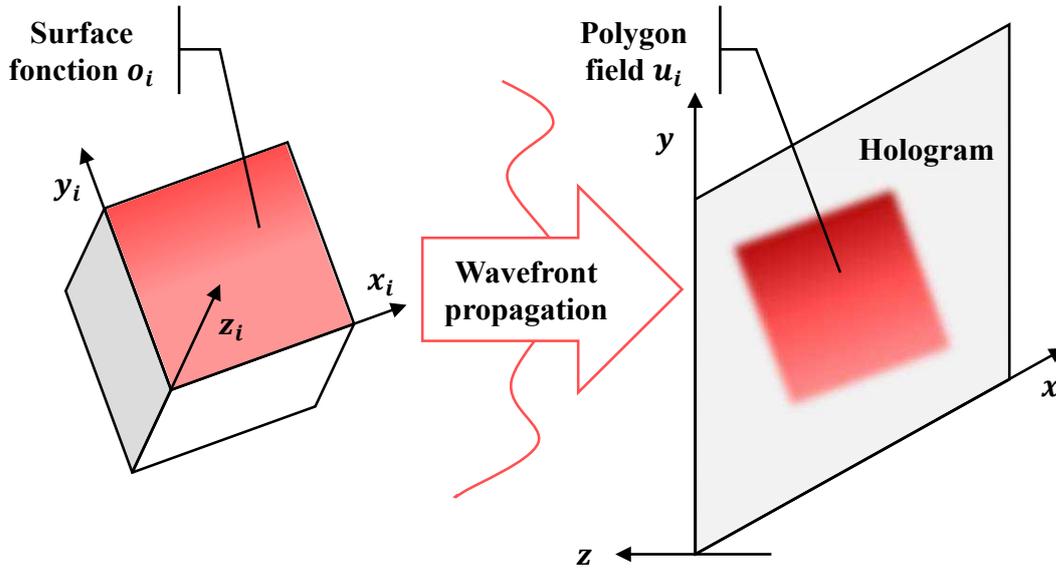


Fig. 3.8 Wave-field approach based on a polygonal modeling of the scene.

between two consecutive layers must not exceed a maximum value [10]. As a consequence, the number of depth layers must be very large if the scene has a large depth extent, increasing the object wave computation time.

To overcome this limitation, a wave-field approach based on a polygonal modeling of the scene has been proposed [99]. In this approach, 3D scenes are described as a set of oriented polygons, where each polygon is considered to be a surface source of light, as shown in Figure 3.8. The global coordinate system is defined by  $(x, y, z)$  so that the hologram lies on the  $(x, y, 0)$  plane. For each polygon  $i$ , a local coordinate system is defined by  $(x_i, y_i, z_i)$  so that the polygon lies on the  $(x_i, y_i, 0)$  plane. Moreover, each polygon is given a surface function  $o_i$  such that

$$o_i(x_i, y_i) = A_i(x_i, y_i) \exp[j\phi_i(x_i, y_i)], \quad (3.8)$$

where the amplitude  $A_i$  gives its shape and texture to the polygon and the phase  $\phi_i$  controls its diffuseness.

The complex light field  $u_i$  emitted by the polygon in the hologram plane is called polygon field. This field cannot be calculated using conventional propagation formulas presented in Section 2.4 because these formulas can only be applied in the case of diffraction between two parallel planes. To compute light propagation between two tilted planes, several authors propose modifications of the conventional propagation formulas, including Rayleigh-Sommerfeld propagation [31], Angular Spectrum propagation [116, 117, 179, 180], as well as Fresnel [40, 99, 219, 220] and Fraunhofer approximations [43, 137, 144].

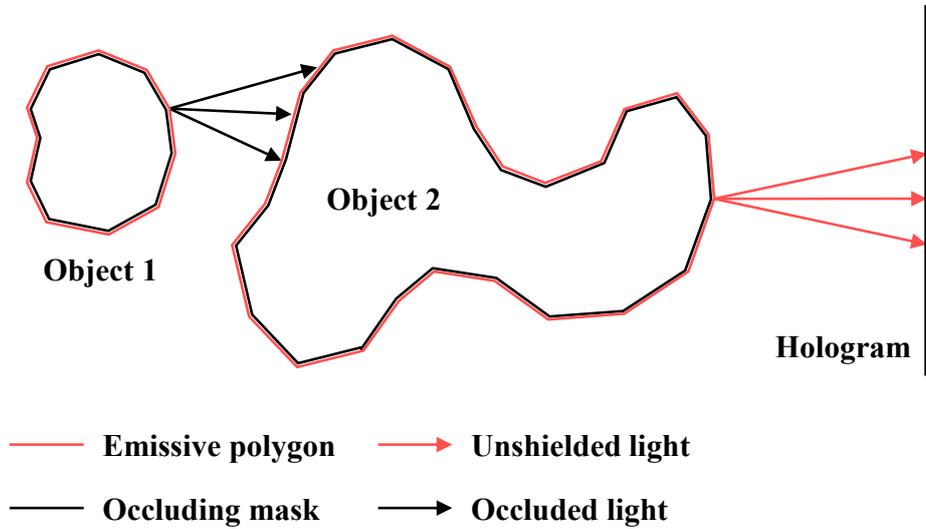


Fig. 3.9 Exact occlusion processing using per-polygon tilted binary masks.

The total object wave can thus be calculated as the sum of polygon fields in the hologram plane, such that

$$O(x, y) = \sum_{i=0}^{M-1} u_i(x, y), \quad (3.9)$$

where  $M$  is the number of polygons in the scene.

### Occlusion processing and shading

Despite its many attractive features, the original method proposed in [99] does not handle occlusions nor surface shading, reducing the realism of the reconstructed object light wave.

In [112], the author extends the occlusion processing algorithm used in [105] to polygon meshes. In this technique, the scene polygons are sorted by depth order and light is propagated from one polygon to another until it reaches the hologram plane. For each polygon, the light wave is shielded by multiplying the field by a tilted binary mask function that has value 0 inside the polygon and 1 elsewhere, as shown in Figure 3.9. Although this occlusion processing technique yields very accurate results and does not produce any visible artifact, its computational complexity is very high, since it requires the field to be propagated from one polygon to another.

In [90, 113, 115], the authors propose a simplified occlusion processing algorithm using per-polygon silhouette binary masks parallel to the hologram plane, as shown in Figure 3.10. Since the propagation of light only needs to be performed from one parallel plane to another, rather than time-expensive propagation between tilted polygons, the computation time is

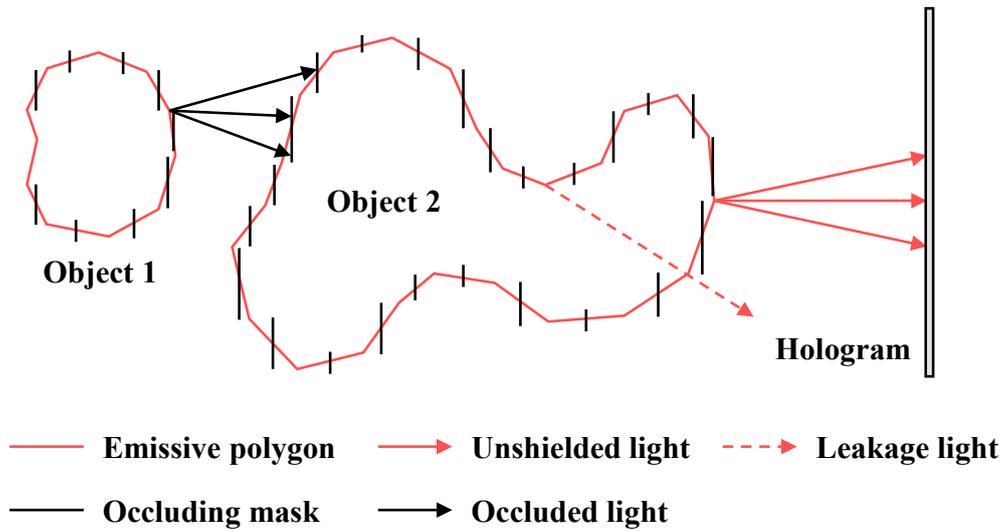


Fig. 3.10 Simplified occlusion processing using per-polygon silhouette binary masks.

greatly reduced. However when the number of scene polygons is large, this technique still requires an enormous computation time, especially for high-resolution CGHs. Moreover, this approximation of the occlusion effect yields leakage light passing through the gaps between silhouette masks, as shown in Figure 3.10.

In [114, 125], the authors further simplify occlusion processing using per-object silhouette binary masks parallel to the hologram plane, as shown in Figure 3.11. For each scene object, a plane is defined such that the cross-section of the object approximately reaches a maximum in it. Then, light is propagated from one object plane to another until it reaches the hologram plane. For each object plane, the light wave is shielded by multiplying the field by a binary silhouette mask that has value 0 inside the orthogonal projection of the object and 1 elsewhere. Since occlusion processing is performed only once per scene object, the object wave computation time is dramatically reduced. However, if an object has a complicated shape or concave surface, leakage light and shadow light are more significant, yielding to black shadows and see-through portions.

Finally, like the point-source approach, CG shading techniques can be integrated during CGH computation to produce photo-realistic effects, such as Phong shading [23, 125, 135, 139, 152], Cook-Torrance reflection [25, 151], Blinn and Torrance-Sparrow models [14, 181, 199–201] and Finite-difference time-domain methods [60, 61, 211]. However, this requires the input scene to be accurately modeled, with the position of light sources and materials composing each scene object fully described. As a consequence, these methods cannot be easily applied to real existing scenes.

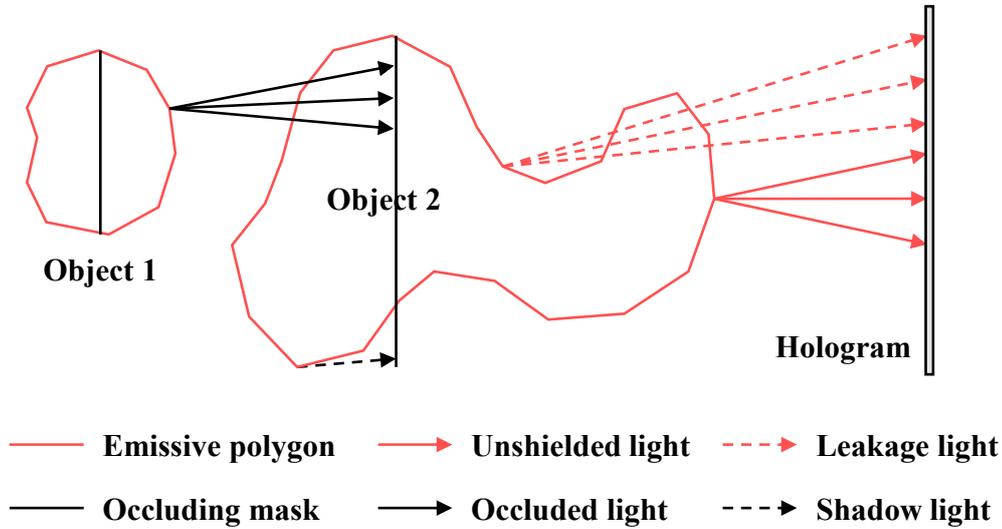


Fig. 3.11 Simplified occlusion processing using per-object silhouette binary masks.

### Computation time reduction

The original wave-field method based on a polygon-meshed representation of the scene [99] requires one to three Fourier transforms per scene polygon. Its computational complexity is therefore in the order of  $\beta MN \log(N)$ , where  $\beta$  is the number of Fourier transforms,  $M$  is the number of scene polygons and  $N = (N_x \times N_y)$  is the number of hologram pixels.

To reduce the object wave calculation time, several methods have been proposed. In [6, 63, 77, 136], the authors propose to analytically compute the object wave of a polygonal modeled scene directly in the frequency domain. The algorithm allows for direct computation of the Fourier spectrum of a general polygon field in the hologram plane analytically, thus eliminating the need to perform any Fourier transform per polygon. Once the whole Fourier spectrum has been calculated in the hologram plane, it is enough to perform one single inverse Fourier transform to obtain the complete object wave. The computational complexity of this approach is in the order of  $\alpha MN + N \log(N)$ , where  $\alpha$  is the number of arithmetic operations,  $M$  is the number of scene polygons and  $N = (N_x \times N_y)$  is the number of hologram pixels. Similarly, in [103, 104], the authors propose to compute the object wave using an analytic formulation of the Fresnel propagation of light scattered by an arbitrarily shaped polygon into the hologram plane without the need for any Fourier transform.

In [150, 154], the authors further reduce the object wave computation time using a LUT approach. The method comprises four steps. First, the polygon field scattered by a basic polygon with fixed shape, size, and orientation is pre-computed in the hologram plane and stored into a LUT. Then, for each scene polygon, the pre-computed polygon field is

transformed in accordance with its translation, rotation, scale size, and skew without the need to perform any Fourier transform. Finally, the complete object wave is computed by summing up the polygon fields calculated in the previous step. The computational complexity of this method is in the order of  $\alpha MN$ , where  $\alpha$  is the number of arithmetic operations,  $M$  is the number of scene polygons and  $N = (N_x \times N_y)$  is the number of hologram pixels. Although this method considerably accelerates the object wave computation, its greatest drawback is the enormous memory size of the LUT required to store the pre-computed polygon field. Methods that use cylindrical [153, 155] or spherical [57] pre-computed polygon fields have been proposed to reduce the memory usage of the LUT.

While these methods allow the object wave computation time to be dramatically reduced, they impose the amplitude to be uniform inside each scene polygon, resulting in a flat shading of the 3D scene. An obvious solution would be to use an increased number of polygons with smaller size and different amplitude values, but it would increase the computational load. In [135], the authors propose an extension of the fully-analytic approach [6] for continuous shading of the scene. In this method, the amplitude distribution within each polygon is given by interpolation according to the distances from the vertices, enhancing the visual aspect of the scene without increasing the number of polygons. Finally, in [96], the authors propose a semi-analytic method for texture mapping using a trade-off between texturing quality and computational efficiency.

### 3.2.3 Object wave computation from multiview data

Despite their many advantages, the two approaches mentioned above only work with synthetic 3D models or 2D view projections supplemented with depth maps. To compute the complex object wave scattered by real-existing scenes without the need for depth maps, two approaches are commonly used: the multiple-viewpoint-projection methods and the holographic stereogram approach. These methods consist of two steps. First, 2D projections of a 3D scene are acquired from various viewpoints using conventional digital cameras. Then, the acquired images are digitally processed to yield the object wave scattered by the scene.

#### Multiview capture

The first step in multiview approaches is to capture a set of perspective projections of the scene. The acquisition process can be performed either along the horizontal axis, yielding Horizontal Parallax Only (HPO) holograms, or on a 2D grid, enabling the computation of full-parallax holograms. HPO holograms are thus easier to acquire, but since the motion parallax is restricted to one transverse axis only, the perceived 3D effect is not as natural as

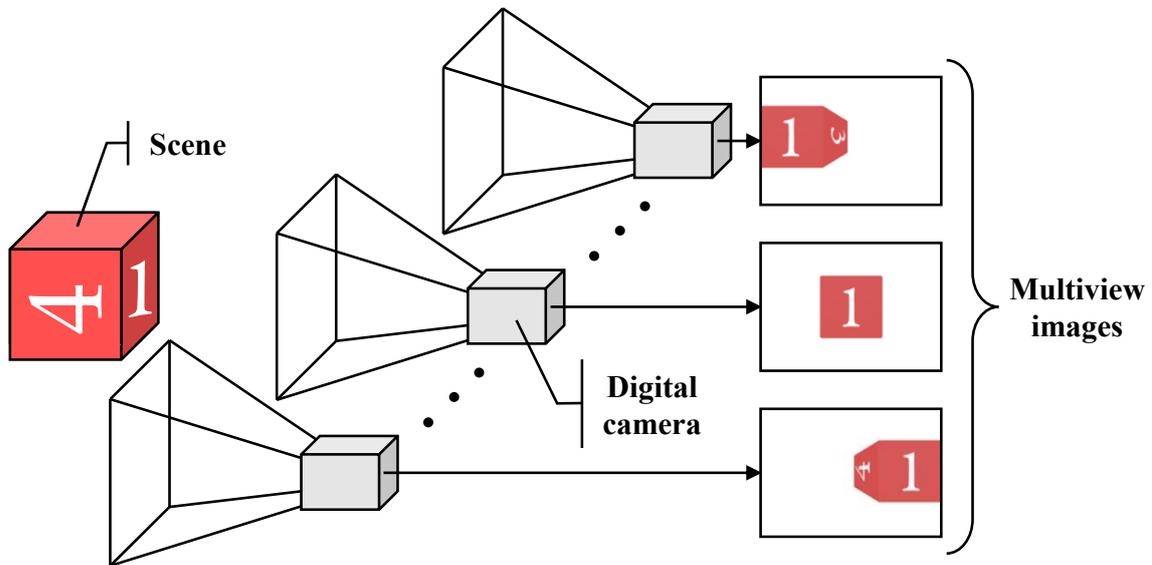


Fig. 3.12 Multiview capture using several cameras arranged along the horizontal axis.

in full parallax holograms. Irrespective of the acquisition axes, the capture process can be performed in several ways.

The most straightforward way to capture the multiview data is to use several cameras arranged along a transverse axis or on a 2D grid [92], as shown in Figure 3.12. By this way, multiple 2D projections of the scene can be acquired simultaneously, enabling the capture of dynamic scenes. However, depending on the number of required views, the camera array can be quite expensive and difficult to set up. To overcome this limitation, it is possible to mechanically shift a single camera along the horizontal and/or vertical axes [100]. The acquisition process can be automated using a step motor to avoid manual handling of the camera. However, the mechanical movement of the camera prevents this method from being used for dynamic scenes.

To reduce the number of required cameras without using any mechanical shift, several methods have been proposed. In [76], the authors use a view synthesis algorithm [156] to generate a series of additional views between consecutive projections by interpolating the locations and intensity values of the corresponding pixels in the given images. In [204], stereo matching is applied to vertical pairs of images to estimate the depth map corresponding to each captured projection. Then, intermediate vertical views are synthesized using Depth-Image-Based Rendering (DIBR) technique with the estimated depth information [226]. In [128, 129], the authors use the Shape-From-Silhouette (SFS) algorithm to generate a voxel model of the scene from the acquired projection images [95]. Then, they synthesize 2D view projections of the scene using volume rendering. The advantage of this method is that

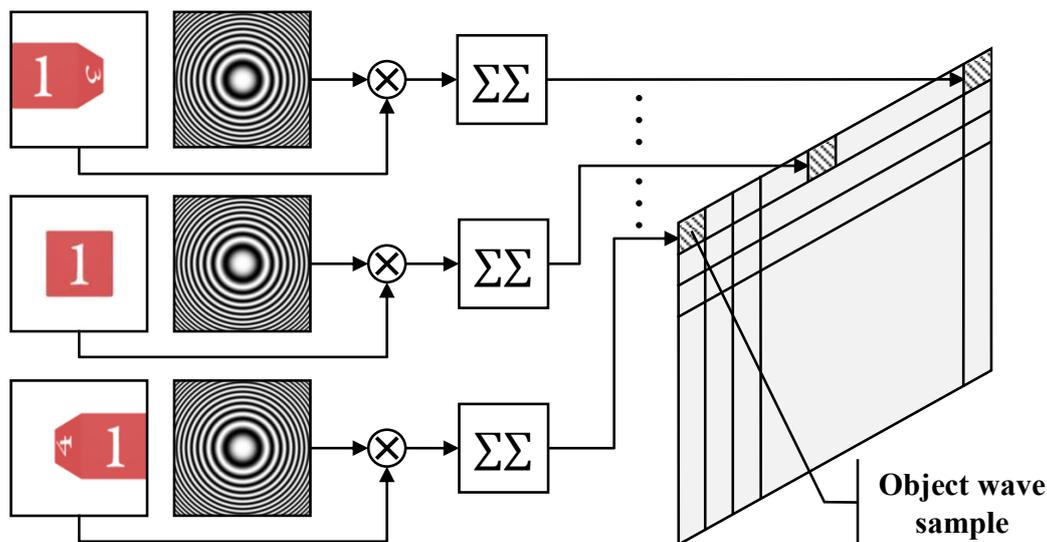


Fig. 3.13 Object wave computation using the multiple-viewpoint-projection approach.

SFS can be performed with just a small number of sparsely arranged cameras. While these methods allow to significantly reduce the number of required cameras, they lead to increased computational complexity.

Alternatively, a microlens array can be used to simultaneously capture a large number of projections in a single camera shot [120, 134, 163], in a similar way as integral imaging systems. Thanks to this method, only one camera is needed to capture the multiview data, without any mechanical shift. The number of acquired projections is given by the number of microlenses within the array, and the resolution of each elemental image is given by the resolution of the camera divided by the number of microlenses. Therefore, this acquisition process leads to a trade-off: having more microlenses in the array allows for the acquisition of more projections with lower resolution. On the other hand, reducing the number of microlenses yields fewer projections with higher resolution. To overcome this limitation, a method which uses macro-lens array together with view synthesis algorithm has been proposed in [159].

### Multiple-viewpoint-projection approach

The first approach for object wave computation from multiview images is the multiple-viewpoint-projection method, proposed in [100] and [2]. In this approach, both HPO and full parallax holograms can be obtained, depending on the multiview acquisition axes [160].

In the case of full parallax holograms, the multiview acquisition must be performed on a transverse 2D grid. We assume that  $(K_x \times K_y)$  view projections of resolution  $(M_x \times M_y)$  are acquired along the horizontal and vertical axes, respectively. The captured images are numbered by  $(x, y)$ , such that the lower-left projection is numbered by  $(0, 0)$  and the upper-right projection by  $(K_x - 1, K_y - 1)$ . To obtain the object wave scattered by the scene, each projection image  $P_{x,y}$  is multiplied by a given Point Spread Function (PSF)  $E_{x,y}$ . Then, the inner product is summed to yield the  $(x, y)$  object wave sample, as shown in Figure 3.13. The object wave is thus given by

$$O(x, y) = \sum_u \sum_v P_{x,y}(u, v) E_{x,y}(u, v), \quad (3.10)$$

where  $E_{x,y}$  depends on the capture set-up used for the multiview acquisition. For instance, in [159, 162], the authors use a quadratic phase function given by

$$E_{x,y}(u, v) = \exp [j2\pi b^2(u^2 + v^2)], \quad (3.11)$$

where  $b$  is an adjustable parameter. The computational complexity of this method is therefore in the order of  $\alpha KM$ , where  $\alpha$  is the number of arithmetic operations,  $K = K_x K_y$  is the number of views and  $M = M_x M_y$  is the number of pixels in each projection image.

In the case of HPO holograms, we assume that  $K_x$  view projections of resolution  $(M_x \times M_y)$  are acquired along the horizontal axis only. The captured images are numbered by  $x$ , such that the left projection is numbered by 0 and the right projection by  $K_x - 1$ . Like full parallax holograms, each projection image  $P_x$  is multiplied by a given PSF  $E_x$  and the inner product of each row is summed up to yield the  $(x, y)$  object wave sample. The object wave is thus given by

$$O(x, y) = \sum_u \sum_v P_x(u, v) E_x(u) \delta_{y,v}, \quad (3.12)$$

where  $E_x$  depends on the capture set-up used for the multiview acquisition and  $\delta_{y,v}$  is the Kronecker delta function. In [161], the authors use a 1D quadratic phase function given by

$$E_x(u) = \exp(j2\pi b^2 u^2), \quad (3.13)$$

where  $b$  is an adjustable parameter. The computational complexity of this method is therefore in the order of  $\alpha K_x M$ , where  $\alpha$  is the number of arithmetic operations and  $M = M_x M_y$  is the number of pixels in each projection image.

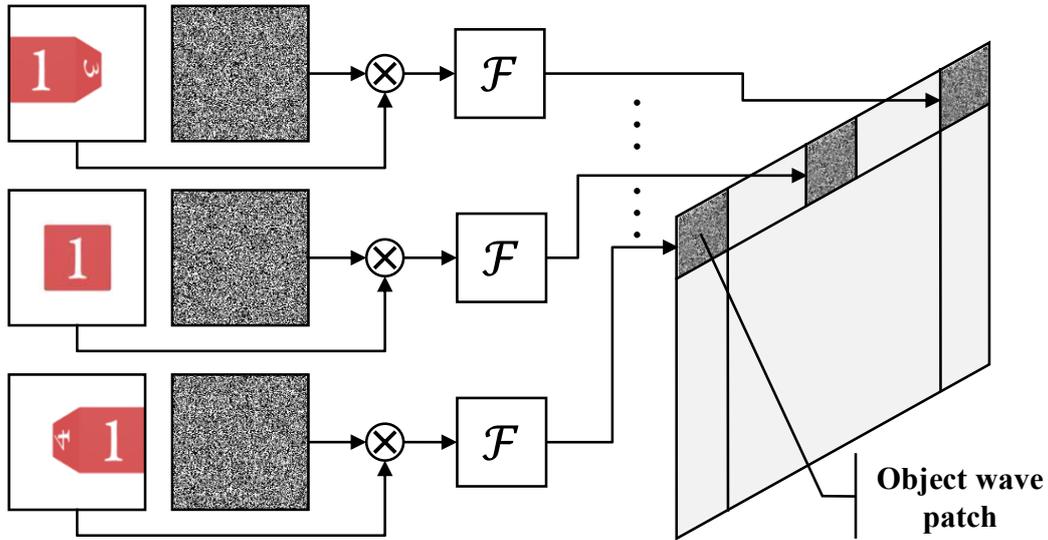


Fig. 3.14 Object wave computation using the holographic stereogram approach.

For HPO as well as full parallax holograms, other PSF functions can be used, such as linear phase functions [3, 18, 19, 66, 78, 133] and random constrained functions [162]. A complete review of multiple-viewpoint-projection methods can be found in [160].

Since occlusions and shading of the scene are naturally recorded in the acquired projections, there is no need to perform any additional calculation to include them in the computed hologram. However, according to Eq. (3.12) and Eq. (3.10), each view projection yields either a single column or a single hologram sample for HPO and full parallax holograms, respectively. As a consequence, the required number of projections is prohibitively huge. For example, to compute the object wave for a hologram resolution of  $(1920 \times 1080)$ , the required number of views is 1920 and 2073600 for HPO and full parallax holograms, respectively. Computing a higher resolution hologram would require a proportionally larger number of images. To overcome this limitation, a method that uses compressive sensing approach [16, 36] has been proposed in [148, 149].

### Holographic stereogram approach

The holographic stereogram is another approach for object wave computation from multiview data, which was first proposed in [210]. In this approach, the hologram is spatially partitioned into several holographic segments, called hogels. The hogels angularly multiplex the 2D projections of the scene towards the corresponding viewing directions. Since the hogels can

be seen as elementary holograms, the object wave patch in each hogel can be computed independently.

We assume that  $(K_x \times K_y)$  view projections of resolution  $(M_x \times M_y)$  are acquired along the horizontal and vertical axes, respectively. The captured images are numbered by  $(m, n)$ , such that the lower-left projection is numbered by  $(0, 0)$  and the upper-right projection by  $(K_x - 1, K_y - 1)$ . For each hogel, the object wave patch is computed by multiplying the corresponding projection image  $P_{m,n}$  by a random phase factor and by calculating the Fourier transform of the inner product, as shown in Figure 3.14. Finally, the object wave patches are stitched together to yield the composite object wave, given by

$$O(x, y) = O(mM_x + u, nM_y + v) = \mathcal{F} \{P_{m,n} \exp(j\phi)\} (u, v), \quad (3.14)$$

where  $\phi \in [0; 2\pi]$  is the uniform random phase. The computational complexity of this method is therefore in the order of  $KM \log(M)$ , where  $K = K_x K_y$  is the number of views and  $M = M_x M_y$  is the number of pixels in each projection image.

This approach is computationally very efficient and does not require as many projection images as multiple-viewpoint-projection methods. Moreover, occlusions and shading of the scene are naturally recorded in the acquired projections, and there is no need to perform any additional calculation to include them in the computed hologram. However, since each hogel only corresponds to a 2D parallax view, holographic stereograms are unable to provide the accommodation cue, which strongly limits the naturalism of the perceived 3D effect.

To improve the depth illusion provided by holographic stereograms, several methods have been proposed, including phase-added stereograms [72–75, 203], integral holograms [214], diffraction-specific stereograms [106], reconfigurable image projection holograms [143], diffraction specific coherent panoramagrams [9, 174], fully computed holographic stereograms [224] and layered holographic stereograms [225]. However, these methods need the 3D geometry of the scene to be described. As a consequence, they only work with synthetic 3D models or 2D view projections supplemented with depth maps.

The other limitation of the conventional holographic stereogram is its poor imaging resolution for scenes with a big depth extent or located far from the hologram plane. Indeed, since each view image used in the computation process is a perspective projection of the scene, objects located far from the cameras plane may be under-sampled, resulting in a blurred image. In [189], the authors propose a Ray-Sampling (RS) plane approach to overcome this limitation. In this method, the projection views are taken or synthesized on a virtual RS plane located near to the 3D scene. Then, the object wave scattered by the scene in the RS plane is calculated using Eq. (3.14) and numerically propagated towards the hologram plane using one of the propagation formulas presented in Section 2.4. This method can be applied to both

Table 3.1 Comparison of object wave computation approaches

<b>Approaches</b>	<b>HVS depth cues</b>	<b>Occlusions</b>	<b>Shading</b>	<b>Remarks</b>
<i>Point-source approach</i>	All	Additional computation	Synthetic scenes only	Efficient for complex shapes
<i>Wave-field approach</i>	All	Additional computation	Synthetic scenes only	Efficient for planar surfaces
<i>Multiple-viewpoint projection</i>	All	Yes	Yes	Very high number of projections
<i>Holographic stereogram</i>	No accomodation	Yes	Yes	High number of projections

virtual and real scenes. Moreover, since the RS plane is located near the scene, the imaging resolution of objects located far from the hologram plane is not degraded. In [190, 191, 202], the authors employ multiple RS planes that are crossing through the scene to compute the object wave of deep 3D scenes. In this method, occlusion processing between objects in the scene is performed based on a conversion between the object wave and light-ray information on each RS plane.

### 3.2.4 Comparison of object wave computation approaches

As stated before, CGH computation techniques should meet several requirements to find application in the field of 3DTV. First, to produce a strong depth illusion, all the human depth cues should be included during object wave computation. Second, to display a realistic 3D scene, occlusions and shading should be accurately reproduced. Finally, to have application in the field of videoconferencing or telepresence systems, object wave computation techniques should be able to produce CGH of natural scenes in real time.

The comparison of object wave computation approaches with respect to these requirements is summarized in Table 3.1. As shown in this table, the point-source and wave-field approaches present the common advantage of being able to produce CGHs with all the human depth cues and occlusion effect. However, to take into account shading of the scene, they require the 3D scene to be accurately modeled, with the position of light sources and materials composing each object fully described. As a consequence, these methods cannot be easily applied to real existing scenes.

Moreover, in the wave-field approach, the object wave computation requires one to three Fourier transforms per scene polygon. The computation of each polygon field is thus more

time-consuming than the computation of the spherical light wave scattered by a single point using the point-source approach. However, complex waves scattered by scene points located within a single polygon are calculated all at once using the wave-field approach. Therefore, this approach is more efficient than the point-source approach when the number of scene points within each polygon is sufficiently important. However, when the scene geometry contains complex shapes, a large number of polygons containing only one or a few points are needed to sample it, making the wave-field approach less efficient than the point-source approach. As a consequence, while the point-source approach is very efficient when the scene contains complex shapes, the wave-field approach is more efficient when objects in the scene consist of large planar surfaces.

As shown in Table 3.1, since occlusions and shading of the scene are naturally recorded in the view projections, there is no need to perform any additional calculation to include them in the computed hologram using the multiple-viewpoint-projection and holographic stereogram approaches. However, since each projection image yields a single hologram pixel in the multiple-viewpoint-projection approach, a huge number of images are required, especially for high-resolution CGHs. On the other hand, holographic stereograms require fewer projection images, but in this case, the accommodation cue is sacrificed, which strongly limits the naturalism of the perceived 3D effect. Moreover, holographic stereograms still require a high number of projections, which are challenging to acquire using a low-cost capture set-up.

Regarding the object wave computation time, several real-time implementations have been demonstrated [64, 75, 94, 192, 207]. However, in these cases the hologram resolution is lower than  $(3840 \times 2160)$ , and the real-time computation of high-resolution CGHs is still very challenging.

### 3.3 Hologram representation

Once the object wave has been calculated in the hologram plane, it is necessary to encode that field into a hologram with real and positive values. Three types of holograms can be obtained from the object wave: amplitude holograms, which modulate the amplitude of the reference wave, phase holograms, which modulate its phase, and complex modulation holograms, which control both the amplitude and phase.

### 3.3.1 Amplitude holograms

Holograms which modulate the amplitude of the incident reference wave are called amplitude holograms. The most straightforward way to compute an amplitude hologram is to numerically simulate the physical phenomena of interference between the object and reference waves occurring in conventional optical holography. Thus, if  $O = |O| \exp(j\phi(O))$  and  $R = |R| \exp(j\phi(R))$  are respectively the object and reference waves in the hologram plane, the computed hologram is given by

$$\begin{aligned} H &= (O + R)(O + R)^* \\ &= |O|^2 + |R|^2 + 2\Re\{OR^*\}, \end{aligned} \quad (3.15)$$

where  $\Re\{C\}$  is the real part of complex number  $C$ . In Eq. (3.15), the first and second terms are the intensities of the object and reference waves, respectively. During image reconstruction, these components are unnecessary and are responsible for the zero diffraction order artifact. The third term is the interference pattern between the object and reference waves. This interference pattern contains all necessary and sufficient holographic information for the object wave restitution.

Since all the holographic information is contained in the third term of Eq. (3.15), the two first terms can be excluded from the computation, yielding

$$H = 2\Re\{OR^*\}, \quad (3.16)$$

as proposed in [107]. Since it contains both positive and negative values, Eq. (3.16) is designated "bipolar intensity". This equation is numerically simpler to calculate than Eq. (3.15) and presents the advantage of not including the zero diffraction order artifact during image reconstruction. Once computed, the hologram must be normalized using scaling and offset operations to have only positive values. This normalization step does not alter the diffraction behavior of the hologram.

### 3.3.2 Phase holograms

Holograms which modulate the phase of the reference wave are called phase holograms. Since they do not modulate the amplitude of the incident wave, phase holograms have better diffraction efficiency than amplitude holograms. However, the calculation of a phase hologram from the complex object wave is a nonlinear ill-posed inverse problem for which analytic solutions cannot be found. Indeed, several different phase holograms may reproduce

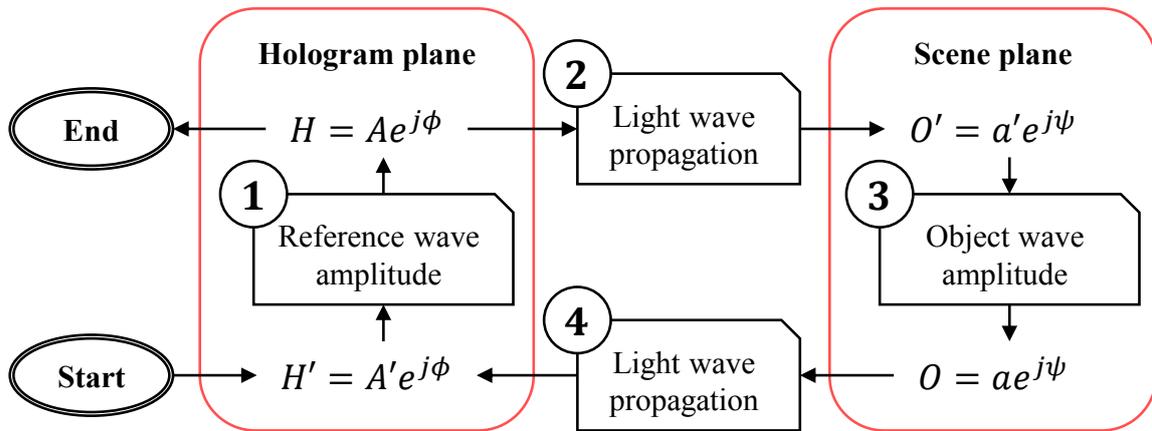


Fig. 3.15 Phase hologram computation using the Gerchberg-Saxton algorithm.

the same object wave. Therefore, the calculation of a phase hologram requires the use of iterative phase retrieval algorithms.

The most popular phase retrieval methods used in CGH are the Iterative Fourier Transform Algorithms (IFTA). The first IFTA, called error-reduction algorithm, has been proposed in [55], and then adapted for phase hologram calculation by Gerchberg and Saxton [45]. The error reduction method is thus often referred as Gerchberg-Saxton (GS) algorithm in the literature. Figure 3.15 shows the block diagram of this algorithm.

The initial hologram pattern is defined to be equal to the computed object wave in the hologram plane. At each iteration, the algorithm then follows these steps:

1. The hologram amplitude  $A'$  is replaced by the amplitude  $A$  of the reference wave used during the reconstruction step, producing a new hologram pattern  $H = A \exp(j\phi)$
2. The light wave is propagated from the hologram plane to the scene plane, yielding the object wave  $O' = a' \exp(j\psi)$
3. The object wave amplitude  $a'$  in the scene plane is replaced by the target scene points amplitude  $a$ , producing a new object wave  $O = a \exp(j\psi)$
4. This new object wave is back-propagated to the hologram plane, producing the hologram pattern  $H' = A' \exp(j\phi)$

This process is then repeated until an acceptable phase hologram which correctly reproduces the object wave has been reached. In [39], the author shows that this algorithm succeeds in reducing the object wave error from one iteration to another.

Other iterative methods may also be used to compute phase holograms, including Simulated Annealing (SA) [88, 218], Direct Binary Search (DBS) [22, 65, 158, 229], Iterative Interlacing Technique (IIT) [17, 38], and Genetic Algorithms (GA) [54, 69, 164, 217].

### 3.3.3 Complex modulation holograms

Holograms which modulate the amplitude and phase of the reference wave are called complex modulation holograms. By controlling both the amplitude and phase of the incident wave, the hologram reproduces the object wave accurately without any unwanted artifact, allowing the viewer to perceive the scene clearly.

Unfortunately, current SLMs can either modulate the amplitude or the phase of an incident wave, but not both. As a consequence, the display of a complex modulation hologram requires two SLMs, one controlling the amplitude of the reference wave, the other modulating its phase [58, 124, 182]. Another method is to use a beam splitter to combine the light waves diffracted by two SLMs, one displaying the real part and the other the imaginary part of the hologram [86, 183, 184]. Finally, other methods have been proposed to display a complex modulation hologram with a single SLM using multiple pixels for each value of the hologram [7, 8, 11, 102]. However, in this case, the resolution is reduced and the hologram must be recalculated to fit the characteristics of the display.

## 3.4 Conclusion

In this chapter, we presented a state of the art of hologram synthesis methods. We have seen that hologram computation generally consists of two steps: (1) calculation of the object wave scattered by the scene into the hologram plane, and (2) representation of this complex wave into a hologram with real and positive values. We presented three different approaches for object wave calculation, each using a different sampling of the scene: the first one is based on a point cloud sampling, the second on a polygon mesh representation, and the third one on multiview data capture. Once calculated, the object wave must be encoded into a real-valued hologram which can modulate the amplitude and/or the phase of the reference wave to reproduce a 3D image of the scene.

Using these methods, it is possible to compute the hologram of a synthetic or real scene without physical interference between light waves. The main limitations associated with conventional optical holography, such as the need for a coherent laser source and highly stable optical system, can be avoided. Real existing scenes can be captured in natural light using multiview data supplemented or not with depth maps, opening the scope of holography

to outdoor 3D video. Many practical applications, such as videoconferencing or telepresence systems, can benefit from the attractive 3D visualization properties of holography.

However, due to the huge amount of data to process, real-time computation of high-resolution CGHs is still very challenging. As a consequence, most CGH computation methods make a trade-off between 3D scene realism, depth accuracy and calculation time. For example, to speed up the computation, some CGHs do not provide all the human depth cues or use a simplified shading of the scene. The objective of this thesis is therefore to propose new algorithms able to produce realistic CGHs with a strong depth illusion and reduced computational complexity.



## **Part III**

### **Hologram calculation time reduction**



# Chapter 4

## Fast hologram computation using a hybrid point-source/wave-field approach

As stated in Chapter 3, it is still very challenging for state-of-the-art methods to compute high-resolution CGHs of 3D scenes at interactive rates. The first part of this thesis thus aims at reducing the object wave calculation time to get closer to real-time computation. To this end, we first designed a novel hybrid method by combining two approaches which complement one another: the point-source and wave-field approaches.

Indeed, as mentioned in Section 3.2.4, while the point-source approach is very efficient when the scene contains complex shapes, such as trees or human bodies, the wave-field approach is more efficient when objects in the scene consist of large planar surfaces, such as roads or buildings. However, most real 3D scenes contain both complex shapes and planar surfaces. As a consequence, these two approaches are rarely fully efficient for computing CGHs of real scenes in their entirety.

To overcome these limitations, we propose a fast CGH computation method based on a hybrid point-source/wave-field approach. Whereas previously proposed methods aim at reducing the computational complexity of the point-source or the wave-field approaches independently, our method uses the two approaches together and therefore takes advantages from both of them. This chapter is organized as follows. Section 4.1 presents the proposed method, Section 4.2 gives a detailed description of the GPU implementation, and the experimental results with respect to various recording parameters are analyzed in Section 4.3.

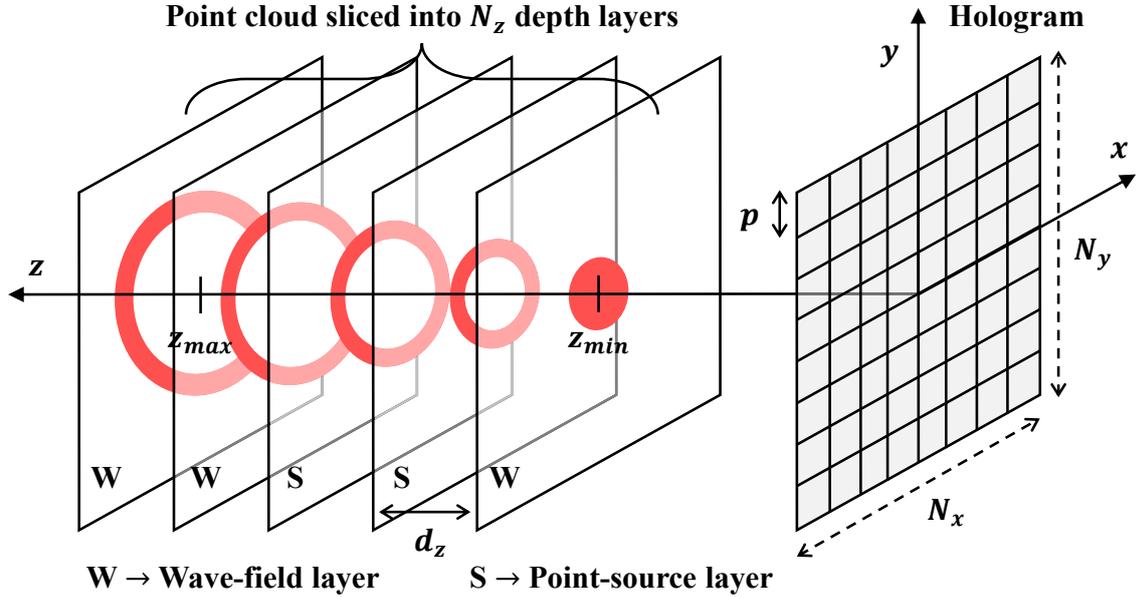


Fig. 4.1 Scene geometry and coordinate system used by the proposed method.

## 4.1 Proposed method

### 4.1.1 Overview

Figure 4.1 shows the scene geometry and coordinate system used by the proposed method. The coordinate system is defined by  $(x, y, z)$  so that the hologram lies on the  $(x, y, 0)$  plane. The 3D scene is sliced into a set of  $N_z$  depth layers parallel to the hologram plane and regularly located between  $z_{min}$  and  $z_{max}$ . We call  $d_z$  the distance between each depth layer.  $N_z$  is set such that the separation between two consecutive layers remains invisible [10]. The depth layers are numbered from 0 to  $N_z - 1$ , from the farthest to the nearest to the hologram plane, and the hologram is sampled on a regular 2D grid of resolution  $(N_x \times N_y)$  with pixel pitch  $p$ .

The depth layers are classified into two categories: the wave-field layers and the point-source layers, as shown in Figure 4.1. This classification is performed depending on the number of scene points within each layer: if the number of scene points  $M_d$  within layer  $d$  exceeds a threshold value  $M_{d,max}$ , this layer is considered to be a wave-field layer, otherwise it is considered to be a point-source layer. Moreover, the farthest and nearest depth layers are always considered as wave-field layers.

Each depth layer  $d$  operates as a surface source of light which emits a complex wave  $o_d$  given by

$$o_d(x, y) = A_d(x, y) \exp [j\phi_d(x, y)], \quad (4.1)$$

where  $A_d(x, y)$  is the amplitude of the  $(x, y)$  point within layer  $d$ , and  $\phi_d(x, y) \in [0; 2\pi[$  is its phase, set to a uniform random value to render a diffuse scene.

Light scattered by each depth layer  $d$  is numerically propagated towards the hologram plane using either a point-source or a wave-field approach depending on which category layer  $d$  belongs to. Then, the complex waves scattered by all the depth layers in the hologram plane are summed up to obtain the object wave  $O$ , such that

$$O(x, y) = O^w(x, y) + O^s(x, y), \quad (4.2)$$

where  $O^w$  and  $O^s$  are respectively the object waves scattered by the wave-field and point-source layers in the hologram plane.

Finally, the CGH  $H$  is computed using the bipolar intensity formula [107], according to

$$H(x, y) = \mathcal{H} \{ \{o_d\}_{0 \leq d < N_z} \} = \Re \{ O(x, y) R^*(x, y) \}, \quad (4.3)$$

where operator  $\mathcal{H}$  stands for CGH computation using the hybrid approach and  $R$  is the reference plane wave.

### 4.1.2 Wave-field layers

When the number of points  $M_d$  within layer  $d$  exceeds  $M_{d,\max}$ , light scattered by this layer is numerically propagated towards the hologram plane using a wave-field approach. To this end, we use the Angular Spectrum propagation, which expresses light propagation between two parallel planes separated by distance  $z_d$  as

$$\mathcal{P}_{z_d}^{AS} \{o_d\} (x, y) = \mathcal{F}^{-1} \left\{ \mathcal{F} \{o_d\} e^{j2\pi z_d \sqrt{\lambda^{-2} - f_x^2 - f_y^2}} \right\} (x, y), \quad (4.4)$$

where  $\lambda$  is the wavelength of light,  $f_x$  and  $f_y$  are the spatial frequencies, and  $\mathcal{F}$  and  $\mathcal{F}^{-1}$  are respectively the forward and inverse Fourier Transform. These transforms can be computed using the Fast Fourier Transform algorithm (FFT).

To avoid one FFT per layer and thus to speed-up the computation, the algorithm sums up the object waves scattered by each wave-field layer directly in the frequency domain and inverse Fourier transforms the result to get  $O^w$ , as proposed in [6]:

$$O^w(x, y) = \mathcal{F}^{-1} \left\{ \sum_{\substack{d=0 \\ M_d > M_{d,\max}}}^{N_z-1} \mathcal{F} \{o_d\} e^{j2\pi z_d \sqrt{\lambda^{-2} - f_x^2 - f_y^2}} \right\} (x, y). \quad (4.5)$$

### 4.1.3 Point-source layers

When the number of points  $M_d$  within layer  $d$  is less than  $M_{d,\max}$ , light scattered by this layer is numerically propagated towards the hologram plane using a point-source approach. Scene points located within the layer are therefore considered as isolated light sources, and light propagation is computed as the sum of light waves scattered by each point. The complex wave scattered by point  $i$  located at coordinates  $(x_i, y_i)$  within layer  $d$  is given by the Angular Spectrum propagation as

$$u_i(x, y) = o_d(x_i, y_i) \mathcal{F}^{-1} \left\{ e^{j2\pi z_d \sqrt{\lambda^{-2} - f_x^2 - f_y^2}} \right\} \otimes \delta(x - x_i, y - y_i), \quad (4.6)$$

where  $o_d(x_i, y_i)$  is the complex amplitude of the point and  $\otimes$  is the convolution operator.

Since convolving a function with a position-shifted Dirac delta shifts it by the same amount, knowing the inverse Fourier transform term in Eq. (4.6) beforehand allows to compute  $u_i$  simply by scaling this term with  $o_d(x_i, y_i)$ , followed by a shifting operation. To speed up the computation, we use a pre-calculated LUT, as proposed in [81]. The LUT  $T$  is pre-computed as

$$T(x, y, z) = \mathcal{F}^{-1} \left\{ e^{j2\pi z \sqrt{\lambda^{-2} - f_x^2 - f_y^2}} \right\} h(x, y, z), \quad (4.7)$$

$h$  being the window function given by Eq. (3.4).

Thanks to its geometrical symmetry and to the window function  $h$ , the LUT needs not be computed for every  $(x, y)$  values. Instead, and to limit its number of samples,  $T$  is pre-computed only within the upper quarter of the square circumscribing the disk defined by  $h$ . Therefore, the number of samples  $N_{T,z}$  of the LUT for depth  $z$  is given by

$$N_{T,z} = \left( \frac{R_{\max}(z)}{p} \right)^2 = \left[ \frac{z}{p} \tan \left( \arcsin \left( \frac{\lambda}{2p} \right) \right) \right]^2, \quad (4.8)$$

where  $R_{\max}$  is given by Eq. (3.3).

Then, the complex wave scattered by point-source layer  $d$  in the hologram plane can be obtained by simply addressing this pre-calculated LUT, such that

$$\mathcal{P}_{z_d}^{PS} \{o_d\} (x, y) = \sum_{i=0}^{M_d-1} o_d(x_i, y_i) T(|x - x_i|, |y - y_i|, z_d), \quad (4.9)$$

where operator  $\mathcal{P}_{z_d}^{PS}$  stands for the numerical propagation of light between two parallel planes separated by a distance  $z_d$  using the point-source approach.

Finally, the object wave scattered by all the point-source layers in the hologram plane is given by

$$O^s(x, y) = \sum_{\substack{d=0 \\ M_d \leq M_{d,\max}}}^{N_z-1} \mathcal{P}_{z_d}^{PS} \{o_d\} (x, y). \quad (4.10)$$

#### 4.1.4 Layers classification

To implement the proposed method, it is necessary to determine the value of  $M_{d,\max}$ . We call  $t^{AS}$  the time needed to compute the complex wave scattered by layer  $d$  at depth  $z_d$  with  $M_d$  luminous points using operator  $\mathcal{P}_{z_d}^{AS}$ , and  $t^{PS}$  the time needed to compute it using operator  $\mathcal{P}_{z_d}^{PS}$ .

Since the computation of  $\mathcal{P}_{z_d}^{AS}$  involves one complex multiplication per pixel and two Fourier transforms,  $t^{AS}$  is only dependent on the number of hologram pixels  $N = (N_x \times N_y)$ . It is given by

$$t^{AS}(N) = \alpha N \log(N) + \beta N, \quad (4.11)$$

where  $\alpha$  and  $\beta$  are constant coefficients.

On the other hand, the computation of  $\mathcal{P}_{z_d}^{PS}$  involves one complex multiplication per sample of the LUT per scene point within the layer, so  $t^{PS}$  is dependent on the number of samples  $N_{T,z_d}$  of the LUT for depth  $z_d$  and on the number of points  $M_d$  within the layer. It is given by

$$t^{PS}(M_d, N_{T,z_d}) = \gamma M_d N_{T,z_d}, \quad (4.12)$$

where  $\gamma$  is a constant. The values of  $\alpha$ ,  $\beta$  and  $\gamma$  are obviously dependent on the implementation and on the computing system used.

To maximize the efficiency of the proposed method,  $M_{d,\max}$  must be set such that

$$t^{AS}(N) = t^{PS}(M_{d,\max}, N_{T,z_d}) \Leftrightarrow M_{d,\max} = \frac{\alpha N \log(N) + \beta N}{\gamma N_{T,z_d}}. \quad (4.13)$$

To find the numerical values of the coefficients  $\alpha$ ,  $\beta$  and  $\gamma$ , we measured the calculation time of the point-source and wave-field approaches for one million randomly generated depth layers with different numbers of scene points. We then found these values using the Gnuplot implementation of the nonlinear least-squares Levenberg-Marquardt algorithm [110].

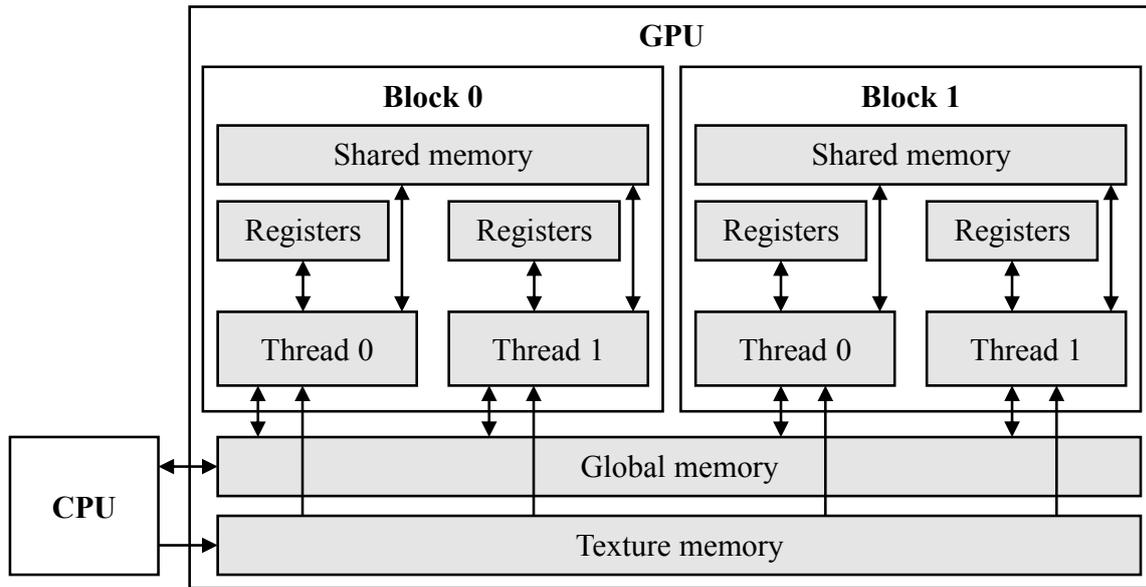


Fig. 4.2 CUDA thread organization and memory model.

## 4.2 Graphics Processing Unit implementation

### 4.2.1 CUDA thread organization and memory model

The proposed method was implemented in C++/CUDA on a PC system employing an Intel Core i7-4930K CPU operating at 3.40 GHz, a main memory of 16 GB, three GPUs NVIDIA GeForce GTX 780Ti, and an operating system of Microsoft Windows 8.

To compute CGH patterns for the three colors simultaneously, we used one CPU thread and one GPU per color. In the implementation, all the calculation is done by the GPUs. The CPU threads are only used to load the input 3D scene, launch CUDA kernels and save the computed CGH into an output file. Finally, to achieve best performance on the GPU, all the computations are performed using single precision.

Figure 4.2 shows the CUDA thread organization and memory model used by the NVIDIA GeForce GTX 780Ti. The parallel portions of an application to be executed on the GPU are called kernels. Each kernel is executed in parallel by many threads, which are organized in blocs. Each thread can independently process and store data using the on-chip registers. Threads within a block can cooperate and exchange data through the on-chip shared memory, which has a short-latency but limited capacity. The CPU and GPU threads can exchange data through the global memory, which has a large capacity but long latency and limited bandwidth. Additionally, the GPU threads can also access to a read-only texture memory,

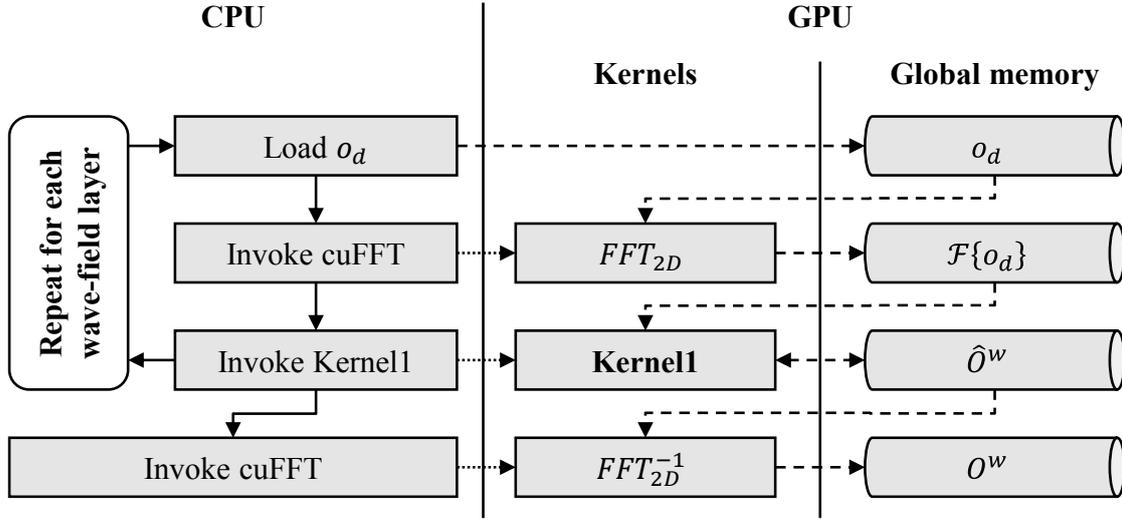


Fig. 4.3 Software structure for calculation of waves scattered by wave-field layers.

which is cached on-chip. The texture cache is optimized for 2D spatial locality, so threads that read texture addresses that are close together will achieve best performance.

## 4.2.2 Wave-field layers

Figure 4.3 shows the software structure for the computation of  $O^w$ . For each wave-field layer  $d$ , the CPU first loads the complex wave  $o_d$  scattered by the layer into the GPU global memory. Then, the Fourier transform in Eq. (4.5) is performed on the GPU using the CUDA cuFFT library by NVIDIA. This library uses the Cooley-Tukey algorithm [26] to optimize the performance of any transform size that can be factored as  $2^a 3^b 5^c 7^d$ , where  $a, b, c$  and  $d$  are non-negative integers. To obtain the best performance, we restrict  $N_x$  and  $N_y$  to be powers of two. Once the Fourier transform is performed, the CPU invokes kernel *Kernel1*, whose pseudo-code is given in Algorithm 1.

---

### Algorithm 1 *Kernel1* pseudo-code

---

**Require:**  $\mathcal{F}\{o_d\}$

**Require:**  $z_d$

**Ensure:**  $\hat{O}^w$

- |  |  |
|--|--|
| <p>1: <b>for all</b> <math>(f_x, f_y)</math> <b>in parallel do</b></p> <p>2:     <math>r \leftarrow \hat{O}^w(f_x, f_y)</math></p> <p>3:     <math>r \leftarrow r + \mathcal{F}\{o_d\} e^{j2\pi z_d \sqrt{\lambda^{-2} - f_x^2 - f_y^2}}</math></p> <p>4:     <math>\hat{O}^w(f_x, f_y) \leftarrow r</math></p> <p>5: <b>end for</b></p> | <p>▷ One CUDA thread per frequency pair</p> <p>▷ Load <math>\hat{O}^w</math> in register</p> <p>▷ Multiply by the transfer function</p> <p>▷ Store <math>\hat{O}^w</math> in global memory</p> |
|--|--|
-

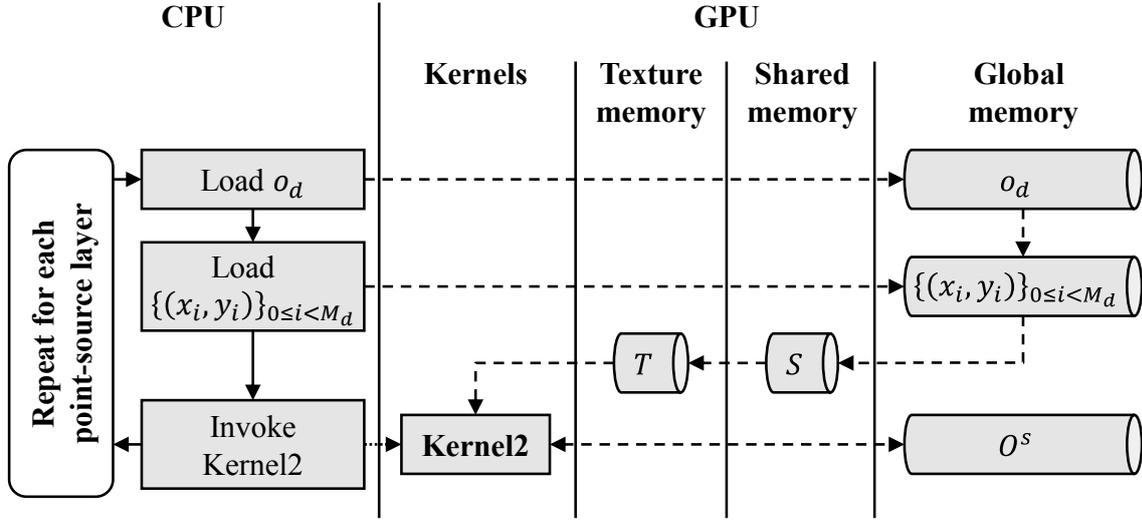


Fig. 4.4 Software structure for calculation of waves scattered by point-source layers.

*Kernel1* simply performs a multiply-accumulate operation on every sample of  $\hat{O}^w$  in parallel, with one thread per sample. Finally, once all the wave-field layers have been processed, the inverse Fourier transform in Eq. (4.5) is performed on the GPU using the cuFFT library.

### 4.2.3 Point-source layers

Figure 4.4 shows the software structure for the computation of complex waves scattered by point-source layers. For each point-source layer  $d$ , the CPU first loads the complex wave  $o_d$  and coordinates  $\{(x_i, y_i)\}_{0 \leq i < M_d}$  of scene points located within layer  $d$  into the GPU global memory. Then, the CPU invokes kernel *Kernel2*, whose pseudo-code is given in Algorithm 2.

*Kernel2* computes the numerical propagation  $\mathcal{P}_{z_d}^{PS} \{o_d\}$  given by Eq. (4.9) on every sample of  $O^s$  in parallel, with one thread per sample. Since the computation of  $\mathcal{P}_{z_d}^{PS} \{o_d\}$  involves one calculation per scene point within layer  $d$  per sample of  $O^s$ , all the threads executing *Kernel2* must access  $o_d$  and  $\{(x_i, y_i)\}_{0 \leq i < M_d}$  data simultaneously. However, this data is stored into the global memory, which has a long latency. Therefore, to reduce and optimize data transfers, *Kernel2* starts out by loading  $\{(x_i, y_i, o_d(x_i, y_i))\}_{0 \leq i < M_d}$  into the shared memory, which has a much shorter latency than global memory. Once this is done, threads must be synchronized to wait until all the data has been properly loaded into shared memory.

*Kernel2* then fetches  $T$  to compute the light waves scattered by each scene point within layer  $d$ . Since all the threads compute the wave scattered by point  $i$  for different samples of

**Algorithm 2** *Kernel2* pseudo-code

---

**Require:**  $o_d$   
**Require:**  $z_d$   
**Require:**  $\{(x_i, y_i)\}_{0 \leq i < M_d}$   
**Ensure:**  $O^s$

```

1: for  $i \in \{0, \dots, M_d - 1\}$  in parallel do                                ▷ One thread per scene point
2:    $x_i^s \leftarrow x_i$                                                     ▷ Load  $x_i$  in shared memory
3:    $y_i^s \leftarrow y_i$                                                     ▷ Load  $y_i$  in shared memory
4:    $o_i^s \leftarrow o_d(x_i, y_i)$                                           ▷ Load  $o_d(x_i, y_i)$  in shared memory
5: end for
6: Synchronize threads
7: for all  $(x, y)$  in parallel do                                          ▷ One thread per object wave sample
8:    $r \leftarrow O^s(x, y)$                                                   ▷ Load  $O^s$  in register
9:   for  $i \in \{0, \dots, M_d - 1\}$  do                                       ▷ For each scene point
10:     $X \leftarrow |x - x_i^s|$ 
11:     $Y \leftarrow |y - y_i^s|$ 
12:    if  $X < R_{\max}(z_d)$  AND  $Y < R_{\max}(z_d)$  then
13:       $r \leftarrow r + o_i^s T(X, Y, z_d)$                                   ▷ Fetch the LUT in texture memory
14:    end if
15:  end for
16:   $O^s(x, y) \leftarrow r$                                                   ▷ Store  $O^s$  in global memory
17: end for

```

---

$O^s$  simultaneously, consecutive threads are expected to fetch adjacent samples of  $T$ . Since the texture memory is optimized for 2D spatial locality memory accesses,  $T$  is stored into this memory to achieve better performance.

Each sample of  $T$  is stored as an 8 bytes single precision complex value in the texture memory. The total amount of memory needed to store  $T$  is therefore given by

$$N_T = 8N_z N_{T, z_{\max}}. \quad (4.14)$$

For  $\lambda = 640\text{nm}$ ,  $p = 8, 1\mu\text{m}$ ,  $N_z = 1024$  and  $z_{\max} = 5\text{cm}$ , the total amount of memory needed to store  $T$  is  $N_T = 488\text{MB}$ . Most current desktop GPUs have more than 512MB of available memory, so this method does not require the use of a professional GPU.

*Kernel2* thus fetches the LUT and performs a multiply-accumulate operation for each scene point  $i$ . To limit the number of accesses to global memory, this multiply-accumulate operation is performed using an on-chip register, and the final result is stored only once into the global memory.

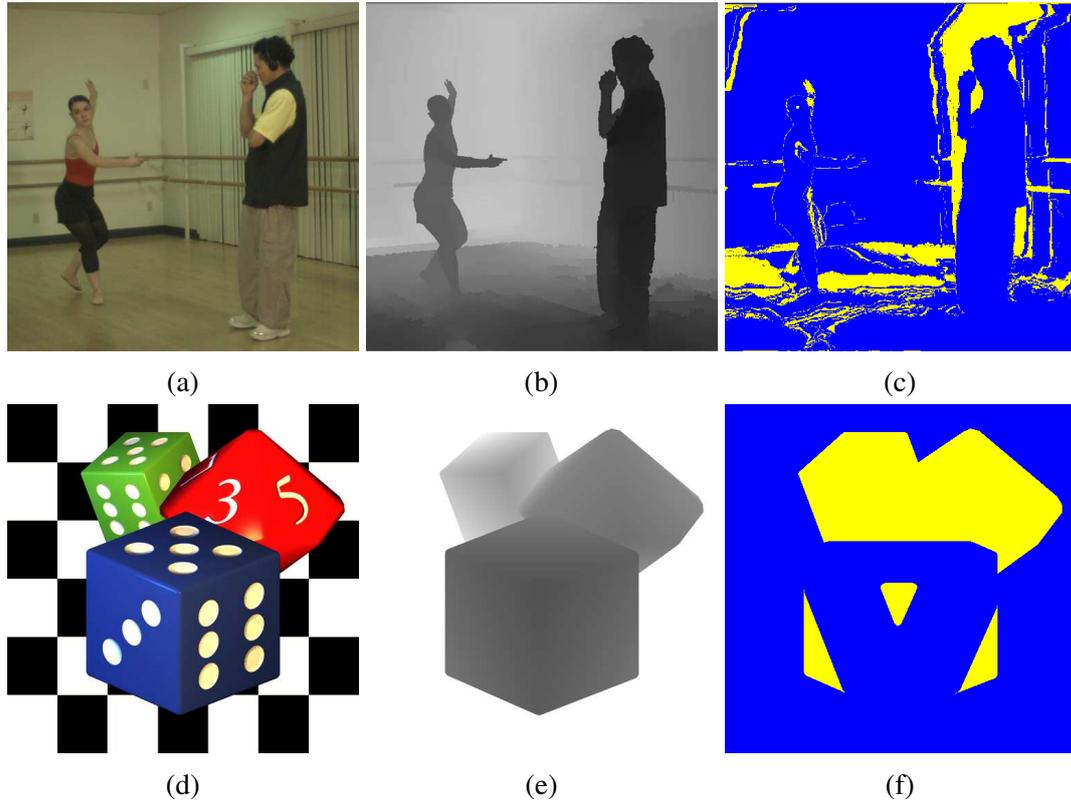


Fig. 4.5 Test scenes used for the experiments: (a), (d) intensity views and (b), (e) depth maps of the scenes *Ballet* and *Dices*. Figures (c) and (f) show in blue the scene points whose complex wave is computed by the proposed method using the wave-field approach and in yellow the scene points whose complex wave is computed using the point-source approach.

## 4.3 Experimental results and discussion

### 4.3.1 Input 3D scenes

For the experiments, we used the views and depth maps of two different tests scenes: the scene *Ballet* from the MSR 3D Video Dataset [230], and *Dices*, a synthetic scene created with Unity3D. The input views and depth maps are shown in Figure 4.5.

From each view and associated depth map, a 3D point cloud is extracted, where each point is given an amplitude proportional to its corresponding pixel value in the view image and a random phase. Since each depth map is encoded as an 8-bits gray level image, the extracted 3D point cloud is naturally sliced as a set of  $N_z = 256$  depth layers parallel to the CGH plane. The total number of points  $M$  within the point cloud is given by the number of pixels ( $M_x \times M_y$ ) of the view and depth map. The 3D point cloud is considered to be located between  $z_{\min} = z_0$  and  $z_{\max} = z_0 + d$  in front of the CGH plane, where  $z_0$  and  $d = 4\text{cm}$

Table 4.1 Hologram synthesis parameters used for the experiments.

Scene parameters		Hologram parameters	
Camera resolution	$(M_x \times M_y)$	Hologram resolution	$(N_x \times N_y)$
Distance of the scene	$z_0$	Pixel pitch	$p = 8.1\mu\text{m}$
Scene depth extent	$d = 4.0\text{cm}$	Red wavelength	640nm
Location of the scene	$(z_0, z_0 + d)$	Green wavelength	532nm
Number of depth layers	$N_z = 256$	Blue wavelength	473nm

are respectively the distance and the depth extent of the scene. Finally, the hologram is sampled on a regular 2D grid of resolution  $(N_x \times N_y)$  with pixel pitch  $p = 8.1\mu\text{m}$ , and the wavelengths are set to 640nm, 532nm and 473nm for the Red, Green and Blue channels, respectively. The hologram synthesis parameters used for the experiments are summarized in Table 4.1.

In the following, we compare the proposed method with GPU implementations of two other methods: (1) the wave-field method proposed in [10], which computes complex wave scattered by each layer using a wave-field approach, and (2) the point-source method proposed in [81], which computes complex wave scattered by each layer using a point-source approach. We adapted both methods to produce colorful CGHs.

### 4.3.2 Analysis of the visual quality of the reconstructed images

We set  $M = 262144$ ,  $N_x = N_y = 4096$  and  $z_0 = 0$ . Figures 4.5c and 4.5f show in blue the scene points whose complex wave is computed by the proposed method using the wave-field approach and in yellow the scene points whose complex wave is computed using the point-source approach. Figures 4.6 and 4.7 show the scene images numerically reconstructed from the CGH patterns of the scenes *Ballet* and *Dices* generated by the point-source method (first column), the wave-field method (second column), and the proposed method (third column). In Figure 4.6, the reconstructed scene images are focused on the man in the first row, and on the dancing girl in the second row. Similarly, in Figure 4.7, the reconstructed scene images are focused on the front dice in the first row, and on the background in the second row. These results show that CGHs computed using the proposed method can provide the accommodation effect as well as using the point-source and wave-field methods. Moreover, when compared to Figures 4.5c and 4.5f, these results show that the proposed method does not produce any visible artifact in the reconstructed images, even at the boundaries between the two categories of points.

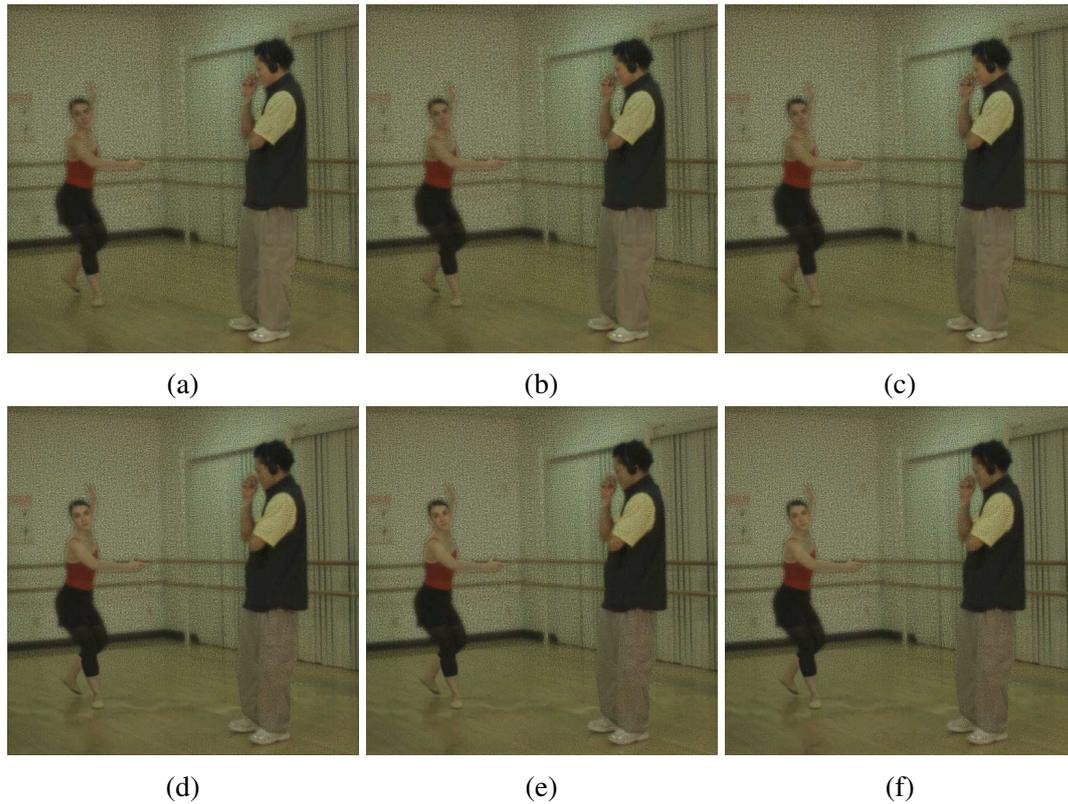


Fig. 4.6 Scene images numerically reconstructed from the CGH patterns of scene *Ballet* generated by the point-source method (first column), the wave-field method (second column), and the proposed method (third column). The reconstructed scene images are focused on the man in the first row, and on the dancing girl in the second row.

To evaluate the objective quality of the reconstructed images compared to the original view image, we used the Peak Signal-to-Noise Ratio (PSNR). The PSNR of the reconstructed images of the scene *Ballet* focused on the man were found to be 20, 26dB, 20, 28dB, and 20, 28dB for the point-source method, the wave-field method and the proposed method, respectively. Similarly, the PSNR of the reconstructed images of the scene *Dices* focused on the front dice were found to be 13, 93dB, 13, 81dB, and 13, 81dB for the point-source method, the wave-field method and the proposed method, respectively. These results show that the proposed method does not reduce the quality of the reconstructed scene images compared to the conventional point-source and wave-field methods. It must be noted that since holograms provide accommodation effect, the numerically reconstructed images have a low depth of field, which is not the case for the original view images. As a consequence, the PSNR of the reconstructed images is found to be below 30dB. Additionally, we compared the CGH patterns generated by the proposed method to those generated by the wave-field and point-source methods using the PSNR. The PSNR of the CGH pattern for the scene

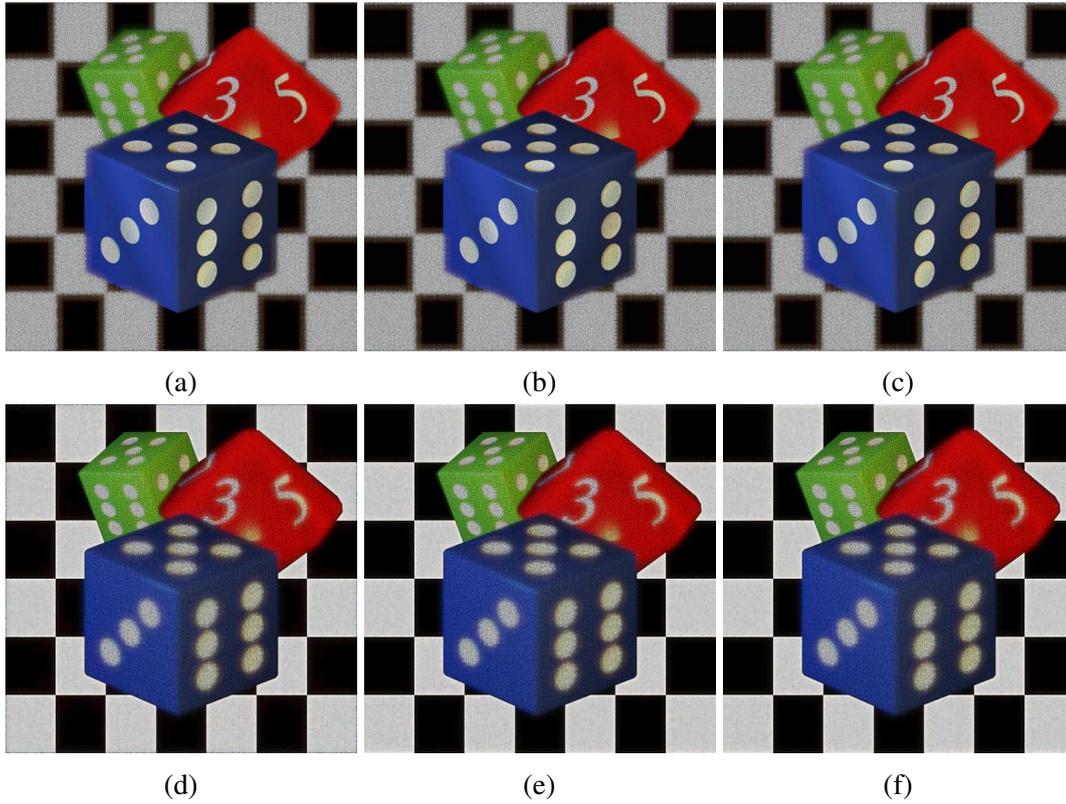


Fig. 4.7 Scene images numerically reconstructed from the CGH patterns of scene *Dices* generated by the point-source method (first column), the wave-field method (second column), and the proposed method (third column). The reconstructed scene images are focused on the front dice in the first row, and on the background in the second row.

*Ballet* generated by the proposed method was found to be 53, 66dB and 57, 52dB, compared to the point-source and wave-field methods, respectively. Similarly, the PSNR of the CGH pattern for the scene *Dices* generated by the proposed method was found to be 54, 73dB and 57, 90dB, compared to the point-source and wave-field methods, respectively. These results show that the proposed method can be used to generate holograms from synthetic scenes as well as real scenes.

### 4.3.3 Analysis of the hologram calculation time

In Figure 4.8, we compare the CGH computation time of the point-source method, the wave-field method and the proposed method depending on the number of scene points  $M$  using views and depth maps of the scene *Dices* with different resolutions. The resolution of the hologram is set to  $(4096 \times 4096)$  and the distance of the scene is set to  $z_0 = 0$ . The number of depth layers remains equal to  $N_z = 256$ , so the average number of scene points

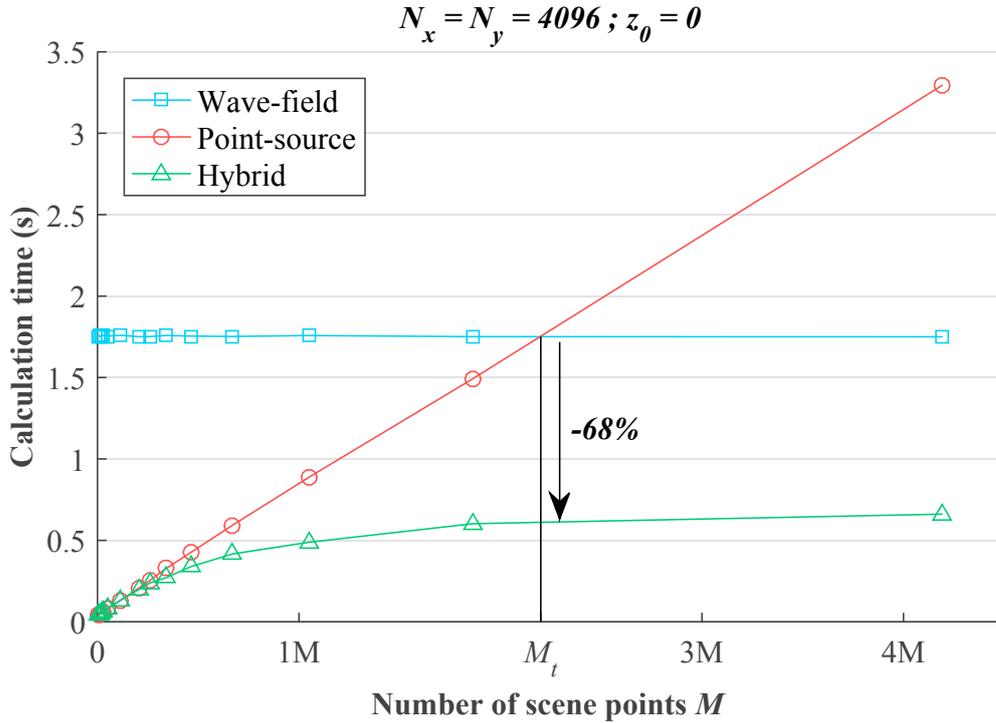


Fig. 4.8 CGH calculation time depending on the number of scene points.

per layer increases with  $M$ . As shown in this figure, while the computation time of the point-source method increases linearly with the number of scene points, the computation time of the wave-field method does not depend on it. Therefore, while the point-source method is faster than the wave-field method for scenes with few points, the wave-field method is still more efficient than the point-source method for scenes with a large number of points. By combining these two approaches, the proposed method takes advantages from both of them and is therefore always the most efficient. The efficiency of the proposed method is maximized when the number of scene points  $M$  reaches a threshold value  $M_t$ , which corresponds to the number of scene points for which the computation time of the point-source method reaches the computation time of the wave-field method. As shown on Figure 4.8, the CGH computation time is reduced by 68% using the proposed method when the number of scene points is equal to  $M_t$ .

Figure 4.9 shows the CGH computation time of the point-source method, the wave-field method and the proposed method for the scene *Dices* depending on the number of hologram pixels  $N$ . The number of scene points is set to  $M = 262144$  and the distance of the scene is set to  $z_0 = 0$ . As shown in this figure, while the computation time of the wave-field method increases linearly with the number of hologram pixels, the computation time of the point-source method does not depend on it. Therefore, while the wave-field method is faster

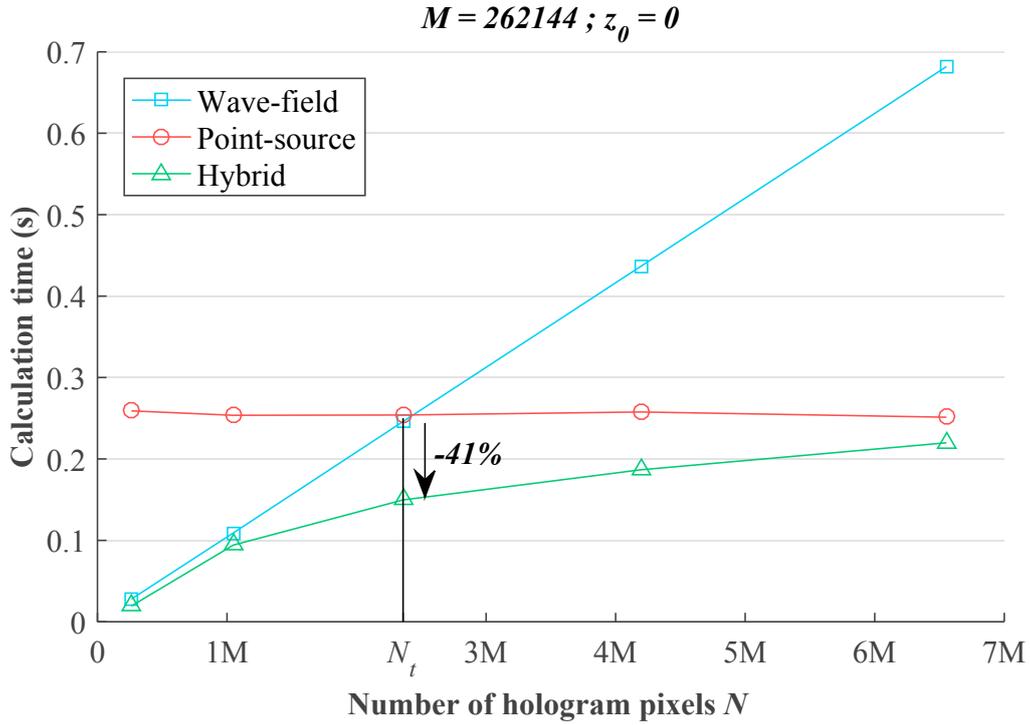


Fig. 4.9 CGH calculation time depending on the number of pixels.

than the point-source method when the number of hologram pixels is low, the point-source method is still more efficient than the wave-field method when the hologram contains a large number of pixels. The proposed method is still the most efficient. The efficiency of the proposed method is maximized when the number of hologram pixels  $N$  reaches a threshold value  $N_t$ , for which the computation times of the wave-field and point-source methods are the same. As shown on Figure 4.9, the CGH computation time is reduced by 41% using the proposed method when the number of hologram pixels is equal to  $N_t$ .

Finally, in Figure 4.10 we compare the CGH computation time of the point-source method, the wave-field method and the proposed method for the scene *Dices* depending on the distance of the scene  $z_0$ . The resolution of the hologram is set to  $(4096 \times 4096)$  and the number of scene points is set to  $M = 262144$ . As shown in this figure, while the computation time of the point-source method increases with the distance of the scene, the computation time of the wave-field method does not depend on it. This is due to the fact that when the distance of the scene increases, the region of contribution of each scene point expands and the number of samples  $N_{T,z_d}$  of the LUT for each depth  $z_d$  increases. Since the time needed to compute the complex wave scattered by a point at depth  $z_d$  is dependent on  $N_{T,z_d}$ , the computation time of the point-source method increases with  $z_0$ . Therefore, while the point-source method is faster than the wave-field method when the scene is located near

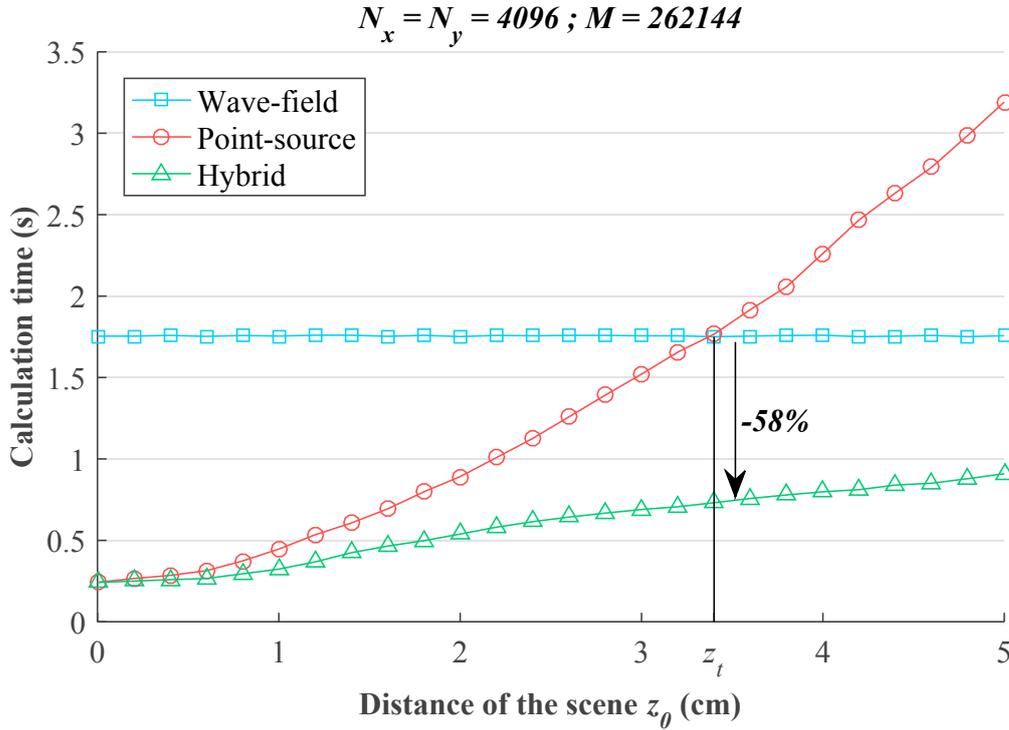


Fig. 4.10 CGH calculation time depending on the distance of the scene.

to the hologram plane, the wave-field method is still more efficient than the point-source method when the scene is located far from the hologram plane. Once again, the proposed method is always the most efficient. The efficiency of the proposed method is maximized when the distance of the scene  $z_0$  reaches a threshold value  $z_t$ , which corresponds to the distance for which the computation times of the point-source and wave-field methods are the same. As shown on Figure 4.10, the CGH computation time is reduced by 58% using the proposed method when the distance of the scene is equal to  $z_t$ .

In addition to the results shown here, we have conducted many tests on both real and synthetic scenes with different values of  $M$ ,  $N$  and  $z_0$ . These tests showed that the proposed method is always more efficient than the point-source and wave-field methods, regardless of the recording parameters  $M$ ,  $N$  and  $z_0$ . Moreover, a reduction of over 65% of the computation time has been reached for each test scene when these parameters were set accordingly. These experimental results confirm the performance superiority of the proposed method over the conventional point-source and wave-field methods in terms of computation time.

## 4.4 Conclusion

In this chapter, we presented our first contribution, dedicated to the reduction of CGH calculation time using a hybrid point-source/wave-field approach. The algorithm consists of three steps. First, the 3D scene is sliced into several depth layers parallel to the hologram plane. Then, for each layer, if the number of points within the layer exceeds a determined maximum value, the complex wave scattered by this layer is computed using a wave-field approach. Otherwise, the complex wave scattered by this layer is computed using a point-source approach. Finally, the complex waves scattered by all the depth layers are summed up to obtain the object wave.

Experimental results revealed that a reduction of over 65% of the computation time has been reached for each test scene compared to the conventional point-source and wave-field methods without producing any visible artifact. This confirms the performance superiority of the proposed method over the conventional point-source and wave-field methods in terms of computation time. However, the calculation of a hologram with  $(2048 \times 2048)$  pixels still takes almost 0.2s for a scene containing 262144 points. In other words, in these conditions the proposed method can only compute 5 CGH frames per second, which is not fast enough for real-time video.



# Chapter 5

## Real-time hologram computation using temporal redundancy removal

As stated in the previous chapter, the hybrid point-source/wave-field method outperforms both point-source and wave-field methods without producing any visible artifact. However, this method can only compute 5 CGH frames of resolution ( $2048 \times 2048$ ) per second, for a scene containing 262144 points, which is not fast enough for real-time video. To further reduce the CGH computation time, it is possible to take advantage of temporal redundancies in a 3D video.

Indeed, there exist only slight changes between consecutive frames in 3D video, and the CGH does not need to be entirely recomputed at each frame. As stated in Chapter 3, several methods have been proposed to track scene changes, including per-point incremental update [85, 142, 177], object-based motion compensation [33, 80], block-based motion compensation [34, 35] and object tracking mask-based method [94]. The principle of these methods is to detect changes in the scene and update the CGH of only affected scene points. Since only small regions of the CGH are updated at each video frame, these methods allow the computational burden to be dramatically reduced. However, all these methods are based on the point-source approach, and the possible exploitation of temporal redundancies using the hybrid point-source/wave-field approach has never been studied.

Accordingly, in this chapter we propose a fast method for the computation of video holograms based on the hybrid approach by taking into account these temporal redundancies. This chapter is organized as follows. The proposed method is first described in Section 5.1, and the experimental results are analyzed in Section 5.2.

## 5.1 Proposed method

### 5.1.1 Overview

The scene geometry and coordinate system used by the proposed method are shown in Figure 4.1. The coordinate system is defined by  $(x, y, z)$  so that the hologram lies on the  $(x, y, 0)$  plane. The input of the method is a 3D video consisting of views and associated depth maps of resolution  $(M_x \times M_y)$ . For each video frame  $i$ , a 3D point cloud is extracted from the view and depth data, where each point is given an amplitude proportional to its corresponding pixel value in the view image. Since each depth map is encoded as an 8-bits gray level image, the extracted 3D point cloud is naturally sliced as a set of  $N_z = 256$  depth layers parallel to the hologram plane and regularly located between  $z_{\min}$  and  $z_{\max}$ . The depth layers are numbered from 0 to  $N_z - 1$ , from the farthest to the nearest to the hologram plane, and the hologram is sampled on a regular 2D grid of resolution  $(N_x \times N_y)$  with pixel pitch  $p$ .

Finally, each layer  $d$  operates as a surface source of light which emits a complex wave  $o_{i,d}$  given by

$$o_{i,d}(x, y) = A_{i,d}(x, y) \exp [j\phi_d(x, y)], \quad (5.1)$$

where  $A_{i,d}(x, y)$  is the amplitude of the  $(x, y)$  point within layer  $d$  for frame  $i$ , and  $\phi_d(x, y) \in [0; 2\pi[$  is its phase, set to a uniform random value to render a diffuse scene.

### 5.1.2 Incremental update

Figure 5.1 shows the overall block-diagram of the proposed method. For each video frame  $i$ , a CGH  $H_i$  of resolution  $(N_x \times N_y)$  with pixel pitch  $p$  is computed. The CGH computation is done differently for the first video frame and for the following frames. For the first frame,  $H_0$  is computed from the view and depth data using the hybrid point-source/wave-field method. For the following frames, the view and depth data are first compared to those of the previous frame to track the differences. A criterion determines whether these differences constitute an important part of the whole scene or not. When the changed part of the scene is important,  $H_i$  is computed from the view and depth data of the current frame  $i$  using the hybrid point-source/wave-field method. Otherwise, if only a small part of the scene has changed, the temporal redundancies can be removed to reduce the computational burden. Therefore, the method computes the CGH of the changed part of the scene only and sums it with the CGH of previous frame  $H_{i-1}$  to get  $H_i$ . Finally,  $H_i$  is transmitted to the CGH video output and stored in the previous frame buffer.

To track the differences between two consecutive video frames  $i - 1$  and  $i$ , the method computes the complex wave  $o'_{i,d}$  scattered by the appearing and disappearing scene points

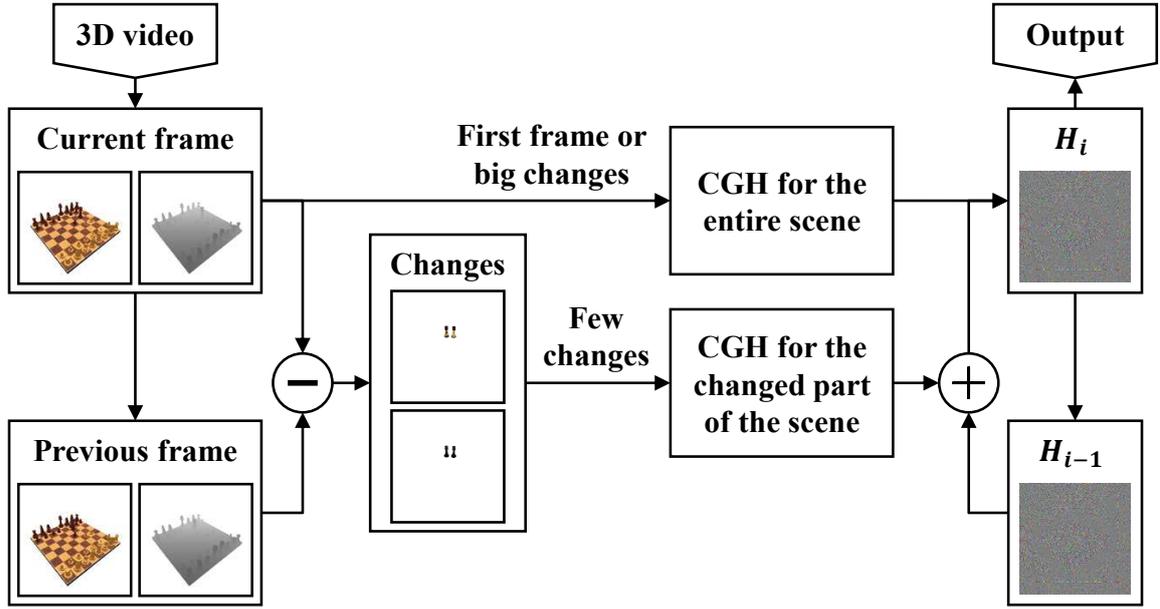


Fig. 5.1 Block-diagram of the proposed method.

within each depth layer  $d$ , given by

$$\begin{aligned} o'_{i,d}(x, y) &= o_{i,d}(x, y) - o_{i-1,d}(x, y) \\ &= (A_{i,d}(x, y) - A_{i-1,d}(x, y)) \exp [j\phi_d(x, y)], \end{aligned} \quad (5.2)$$

where  $A_{i,d}(x, y)$  and  $A_{i-1,d}(x, y)$  are the amplitude of the  $(x, y)$  point within layer  $d$  for frames  $i$  and  $i - 1$ , respectively. The CGH of the changed parts of the scene is therefore given by

$$H'_i(x, y) = \mathcal{H} \left\{ \left\{ o'_{i,d} \right\}_{0 \leq d < N_z} \right\}. \quad (5.3)$$

The CGH  $H_i$  can thus be computed as

$$\begin{aligned} H_i(x, y) &= \mathcal{H} \left\{ \left\{ o_{i,d} \right\}_{0 \leq d < N_z} \right\} \\ &= \mathcal{H} \left\{ \left\{ o'_{i,d} + o_{i-1,d} \right\}_{0 \leq d < N_z} \right\} \\ &= \mathcal{H} \left\{ \left\{ o'_{i,d} \right\}_{0 \leq d < N_z} \right\} + \mathcal{H} \left\{ \left\{ o_{i-1,d} \right\}_{0 \leq d < N_z} \right\} \\ &= H'_i(x, y) + H_{i-1}(x, y). \end{aligned} \quad (5.4)$$

Since there exist only slight changes in view and depth data between two consecutive frames of the 3D video, the number of appearing and disappearing points  $M'_{i,d}$  between frames  $i - 1$  and  $i$  may be far less than the total number of scene points  $M_{i,d}$  for frame  $i$ .

Therefore, the computation of  $H'_i$  for the changed parts of the scene may be faster than the computation of  $H_i$  for the whole scene. The CGH computation time  $t_i$  for frame  $i$  using Eq. (4.3) is given by

$$t_i = \sum_{d=0}^{N_z-1} t^{PS}(\min(M_{i,d}, M_{d,\max}), N_{T,z_d}) \quad (5.5)$$

We call  $t_i$  and  $t'_i$  the computation times of  $H_i$  and  $H'_i$  using Eq. (4.3), respectively. The computation of  $H_i$  using Eq. (5.4) is therefore faster than using Eq. (4.3) if

$$t'_i < t_i$$

$$\sum_{d=0}^{N_z-1} \min(M'_{i,d}, M_{d,\max}) N_{T,z_d} < \sum_{d=0}^{N_z-1} \min(M_{i,d}, M_{d,\max}) N_{T,z_d}. \quad (5.6)$$

For each video frame  $i$ , the CGH  $H_i$  is therefore computed using the recurrence formula

$$H_i(x, y) = \begin{cases} \mathcal{H}\left\{\{o_{i,d}\}_{0 \leq d < N_z}\right\} & \text{if } i = 0 \text{ or } t'_i \geq t_i \\ \mathcal{H}\left\{\{o'_{i,d}\}_{0 \leq d < N_z}\right\} + H_{i-1}(x, y) & \text{otherwise.} \end{cases} \quad (5.7)$$

## 5.2 Experimental results and discussion

The proposed method was implemented in C++/CUDA on a PC system employing an Intel Core i7-4930K CPU operating at 3.40 GHz, a main memory of 16 GB, three GPUs NVIDIA GeForce GTX 780Ti, and an operating system of Microsoft Windows 8.

Table 5.1 shows the hologram parameters used for the experiments. The hologram is sampled on a regular 2D grid of resolution  $(2048 \times 2048)$  with pixel pitch  $p = 8, 1\mu\text{m}$ . The wavelengths are set to 640nm, 532nm and 473nm for the Red, Green and Blue channels, respectively. Finally,  $S_{\max}$  is set to 100.

### 5.2.1 Input 3D video

For the experiments, we generated with Unity an input 3D video containing 375 views and associated depth maps of resolution  $(512 \times 512)$ . From each view and associated depth map, a 3D point cloud is extracted, where each point is given an amplitude proportional to its corresponding pixel value in the view image. The extracted 3D point cloud is considered to be located between  $z_{\min} = 0$  and  $z_{\max} = 2\text{cm}$  in front of the CGH plane.

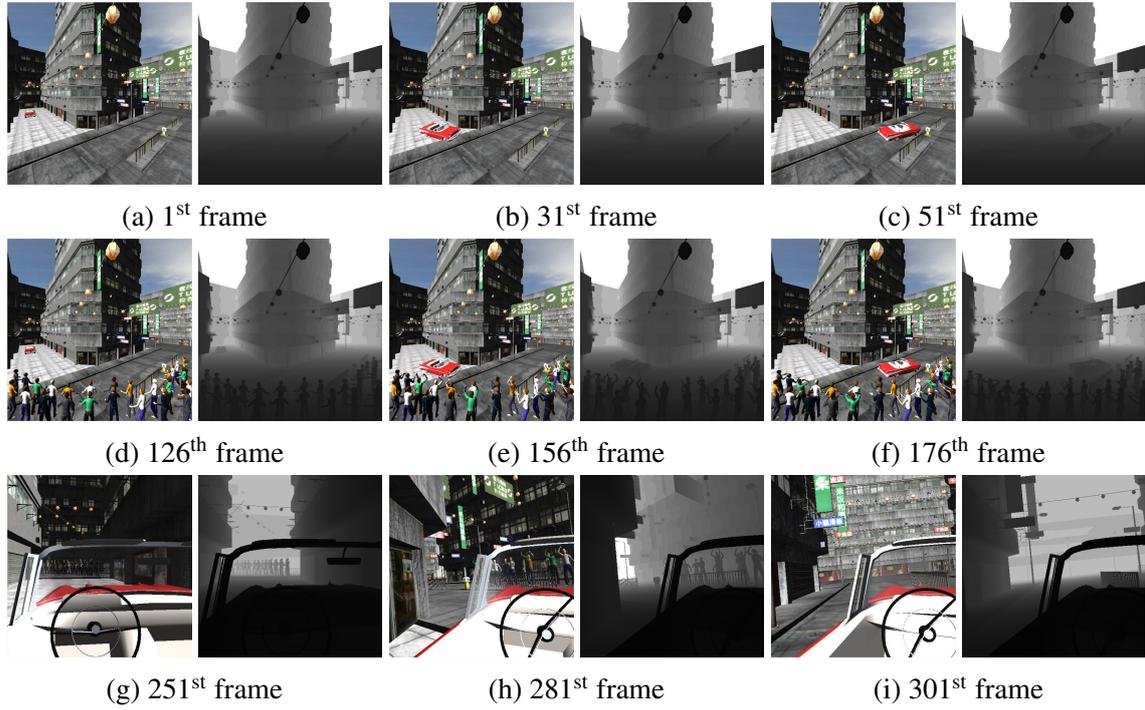


Fig. 5.2 Views and associated depth maps for nine input 3D video frames.

Figure 5.2 shows the views and associated depth maps for nine different frames of the video, which is composed of three parts. The first part (1<sup>st</sup> to 125<sup>th</sup> frames) shows a car turning around a building in a street. The second part (126<sup>th</sup> to 250<sup>th</sup> frames) shows the same scene with some cheering people. Finally, the third part (251<sup>st</sup> to 375<sup>th</sup> frames) shows the same scene from the driver's viewpoint. In the first and second parts of the video, the camera is fixed, whereas it is moving with the car in the third part of the video. As a consequence, while consecutive frames in the first and second parts of the video differ slightly from each other, there exist large changes between consecutive frames in the third part of the video.

Table 5.1 Hologram synthesis parameters used for the experiments.

Scene parameters		Hologram parameters	
Camera resolution	$(512 \times 512)$	Hologram resolution	$(2048 \times 2048)$
Location of the scene	$(0, 2\text{cm})$	Pixel pitch	$p = 8.1\mu\text{m}$
Number of depth layers	$N_z = 256$	Red wavelength	640nm
		Green wavelength	532nm
		Blue wavelength	473nm

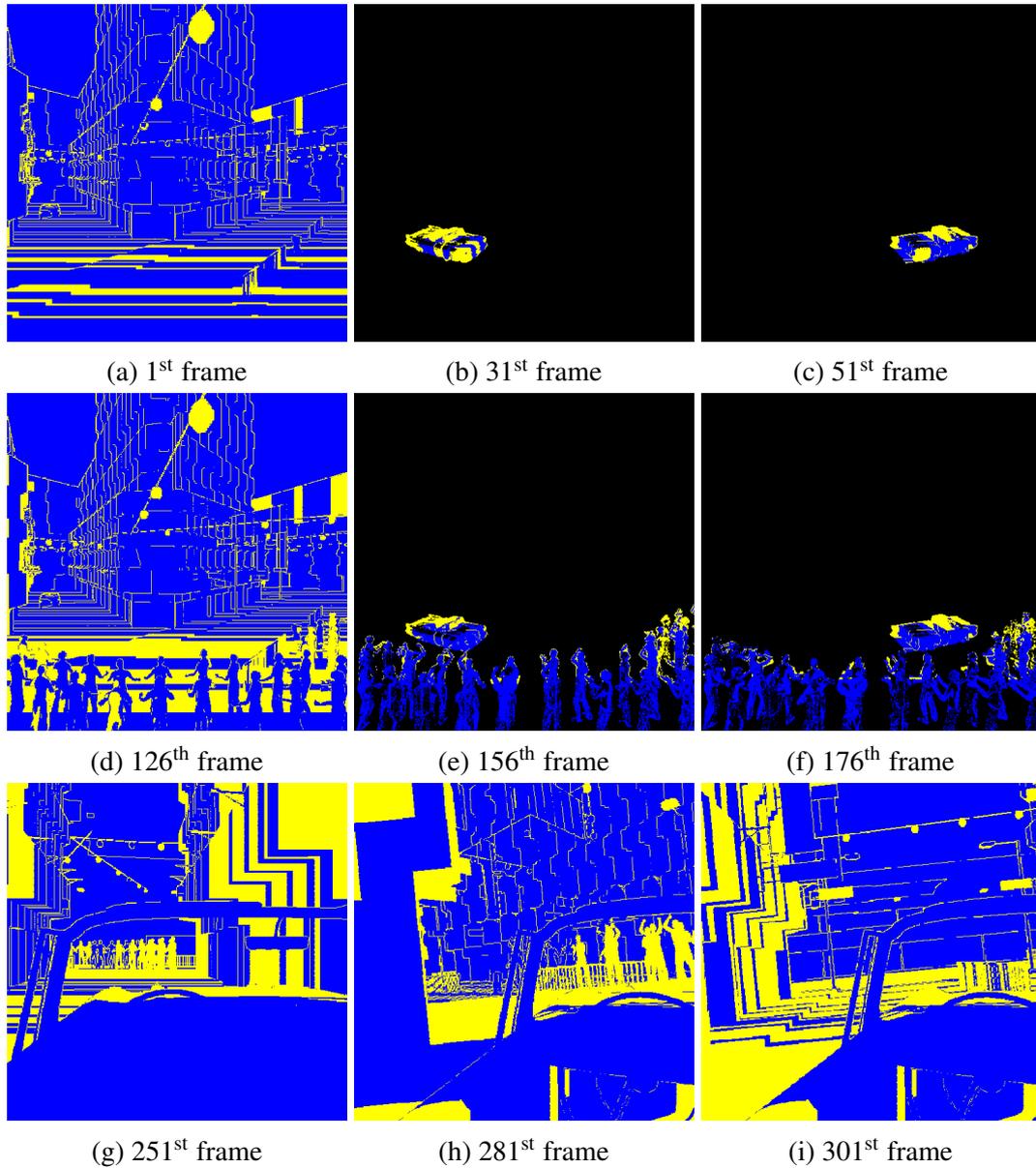


Fig. 5.3 Scene points whose complex waves are updated at each frame. The scene points whose complex waves are computed by the proposed method using the wave-field approach are shown in blue. The scene points whose complex waves are computed using the point-source approach are shown in yellow.

## 5.2.2 Analysis of the visual quality of the reconstructed images

Figure 5.3 shows the scene points whose complex waves are updated at each frame. The scene points whose complex waves are computed using the wave-field approach are shown in blue, whereas those whose complex waves are computed using the point-source approach are shown in yellow. As shown in this figure, since there exist slight changes between

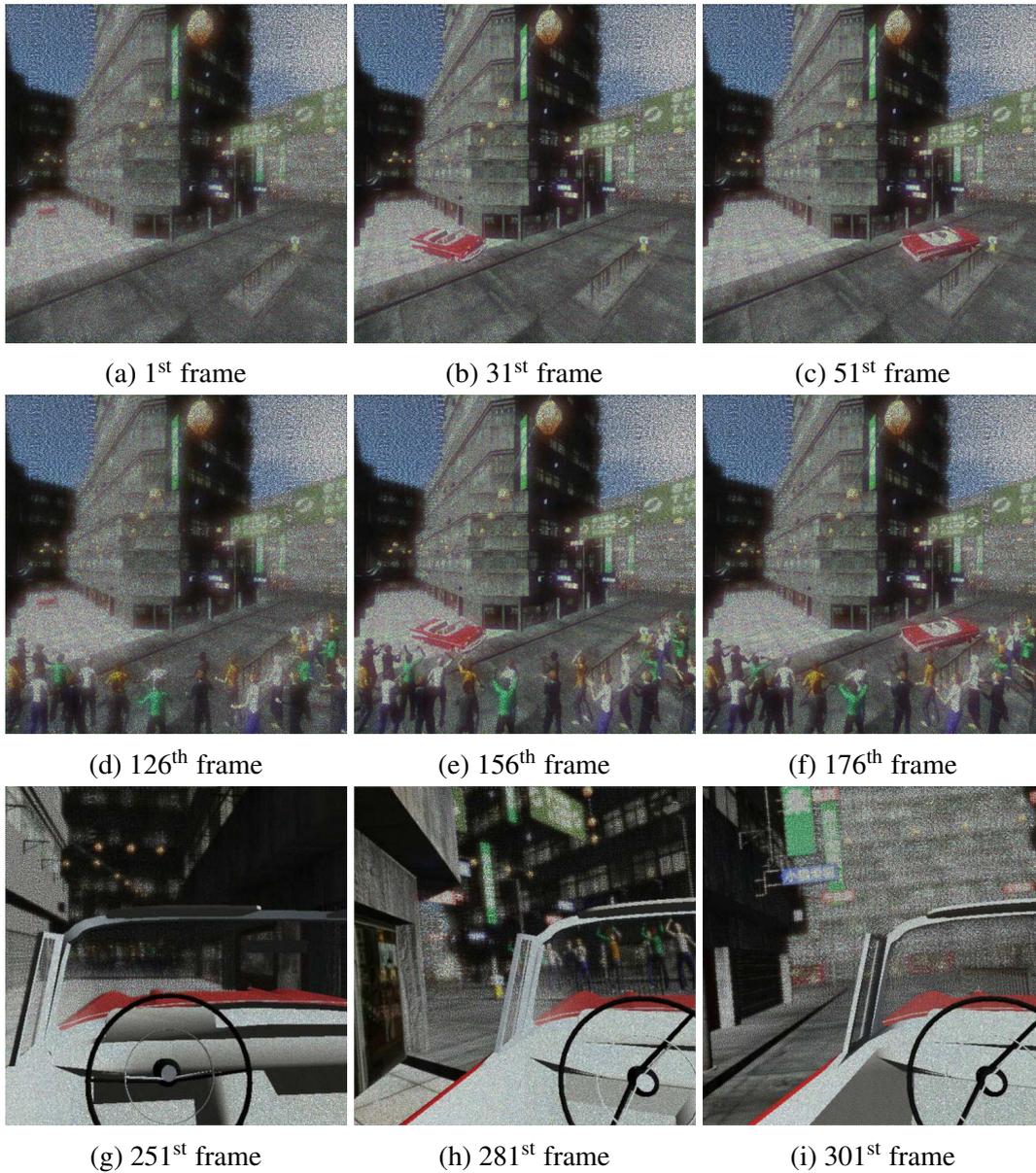


Fig. 5.4 Numerical reconstructions of nine different CGH frames.

consecutive frames in the first and second parts of the video, the proposed method updates the complex waves scattered by only a few scene points at each frame. On the other hand, since a large part of the scene changes between consecutive frames in the third part of the video, the proposed method recomputes the complex wave scattered by the whole scene at each frame. Moreover, due to the sudden changes of the scene between the first and second parts of the video, the proposed method also recomputes the complex wave scattered by the whole scene for the 126<sup>th</sup> frame.

Figure 5.4 shows the scene images numerically reconstructed from the CGH patterns of nine different frames. For the first and second parts of the video, the reconstructed images are focused on the facade of the building, and for the third part of the video, the reconstructed images are focused on the steering wheel. As shown in this figure, the proposed method does not produce any visible artifact, even at the boundaries between the two categories of scene points. To evaluate the objective quality of the reconstructed images compared to the hybrid method presented in Chapter 4, we used the PSNR. The PSNR of the reconstructed images were found to be higher than 50dB for each video frame. These experimental results show that the CGH pattern is accurately updated at each video frame and that the proposed method does not reduce the quality of the reconstructed scene images compared to the hybrid method.

### 5.2.3 Analysis of the hologram calculation time

To evaluate the performance of the proposed method in terms of computation time, we compare it with GPU implementation of four other methods: (1) the method proposed in [10], based on a wave-field approach, (2) the method proposed in [81], based on a point-source approach, (3) the method proposed in [85], based on a point-source approach and incremental update of the CGH at each frame, and (4) the hybrid method presented in Chapter 4. In the following, these methods are called wave-field method, point-source method, incremental method and hybrid method, respectively. We adapted the first three methods to produce colorful CGHs.

Figure 5.5 shows the number of scene points whose complex waves are computed at each frame for all the compared methods. Since the wave-field, point-source and hybrid methods do not take into account temporal redundancies between consecutive frames in the 3D video, the number of calculated scene points remains the same for each video frame. By contrast, the incremental and proposed methods take into account those redundancies to reduce the number of calculated scene points. As a consequence, since consecutive frames in the first and second parts of the video differ slightly from each other, the number of calculated points is much lower for the incremental and proposed methods than for the other methods. Due to the cheering people appearing in the second part of the video, the number of calculated scene points is higher for these frames than for the first part of the video. Finally, since there exist large changes between consecutive frames in the third part of the video, the number of calculated scene points is exactly the same for all the compared methods.

Figure 5.6 shows the CGH computation time per frame for all the compared methods. The computational complexity of the wave-field method is dependent on the number of depth layers containing scene points. By contrast, the computational complexity of the point-source

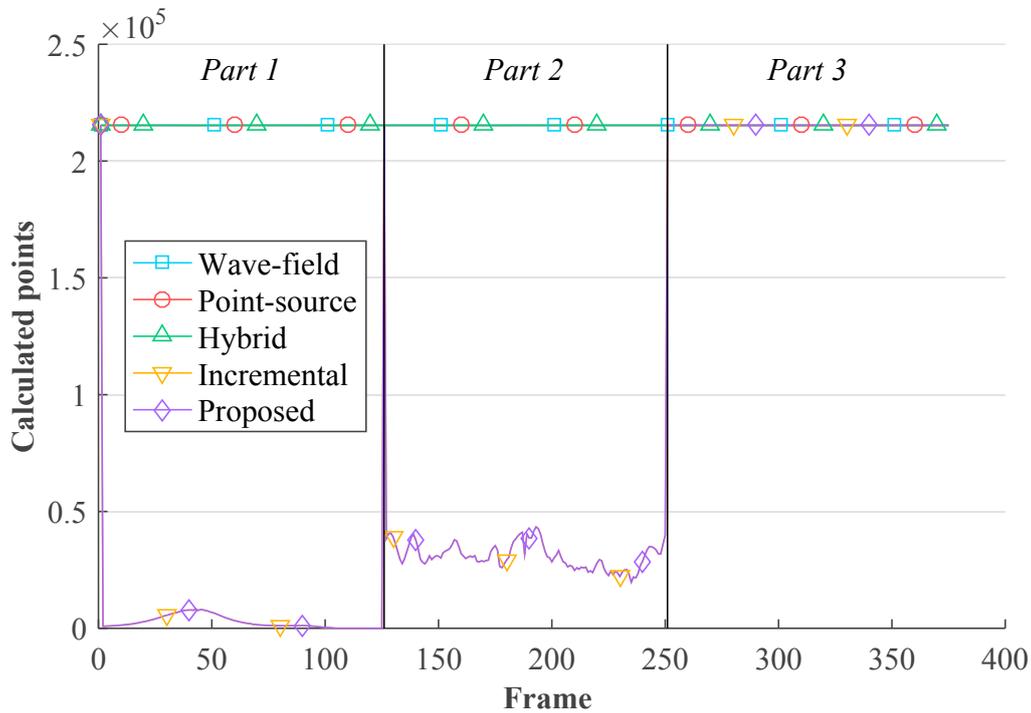


Fig. 5.5 Number of points whose complex waves are calculated for each frame.

method is dependent on the number of scene points and on their location within the scene. Since consecutive frames in the first and second parts of the video differ slightly from each other, the number of non-empty depth layers and the location of scene points are almost the same from one frame to another. As a consequence, the CGH computation times of the wave-field and point-source methods remain almost constant for the first and second parts of the video. On the contrary, since there exist large changes between consecutive frames in the third part of the video, the number of non-empty depth layers and the location of scene points change from one frame to another. Therefore, the CGH computation times of the wave-field and point-source methods also change from one frame to another.

Since it combines the wave-field and point-source approaches, the computational complexity of the hybrid method is dependent on the number of depth layers containing scene points, on the number of scene points and on their location within the scene. As a consequence, its CGH computation time varies in the same way as those of the wave-field and point-source methods, as shown in Figure 5.6. However, by combining these two approaches, the hybrid method takes advantages from both of them and is therefore always more efficient than the wave-field and point-source methods.

In the first and second parts of the video, the number of calculated points for the incremental and proposed methods is much lower than for the other methods. As a consequence,

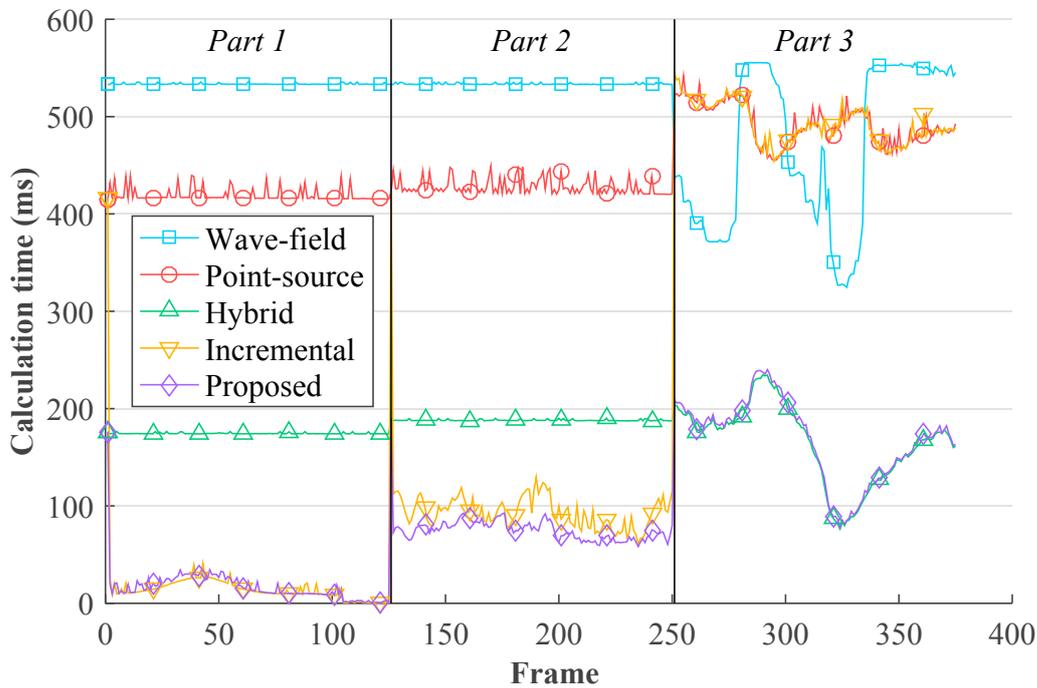


Fig. 5.6 Hologram calculation time for each frame.

the incremental and proposed methods are faster than the wave-field, point-source and hybrid methods. While the CGH computation times of the incremental and proposed methods are almost the same for the first part of the video, the proposed method is more efficient than the incremental method for the second part of the video. Finally, in the third part of the video, the number of calculated scene points is exactly the same for all the compared methods, and the CGH computation times of the incremental and proposed methods reach those of the point-source and hybrid methods, respectively. As a consequence, the proposed method is much more efficient than the incremental method for the third part of the video.

Table 5.2 shows the average number of calculated scene points and the average CGH computation time per frame for the hybrid, incremental and proposed methods. As shown in this table, for the first part of the video, the average number of calculated scene points per frame is equal to 215293 for the hybrid method and to 4282 for the incremental and proposed methods, which means that it has been reduced by 98,01% using the incremental and proposed methods. As a consequence, the average CGH computation time per frame is equal to 174,46ms, 16,69ms and 16,60ms for the hybrid, incremental and proposed methods, respectively. The CGH computation time per frame has thus been reduced by 90,43% and 90,48% using the incremental and proposed methods, respectively. In other

Table 5.2 Average number of calculated points and CGH calculation time per frame.

	<b>Video</b>	<b>Hybrid</b>	<b>Incremental</b>	<b>Proposed</b>
<b>Calculated points</b>	<i>Part 1</i>	215293 (100%)	4282 (1, 99%)	4282 (1, 99%)
	<i>Part 2</i>	215293 (100%)	32537 (15, 1%)	32537 (15, 1%)
	<i>Part 3</i>	215296 (100%)	215296 (100%)	215296 (100%)
<b>Calculation time (ms)</b>	<i>Part 1</i>	174, 46 (100%)	16, 69 (9, 57%)	16, 60 (9, 52%)
	<i>Part 2</i>	188, 02 (100%)	96, 15 (51, 1%)	75, 37 (40, 1%)
	<i>Part 3</i>	164, 57 (100%)	492, 80 (299%)	168, 07 (102%)

words, the proposed method is able to produce colorful CGHs at 60 frames per second, which is sufficient for real-time systems.

Similarly, for the second part of the video, the average number of calculated scene points per frame is equal to 215293 for the hybrid method and to 32537 for the incremental and proposed methods, which means that it has been reduced by 84, 9% using the incremental and proposed methods. As a consequence, the average CGH computation time per frame is equal to 188, 02ms, 96, 15ms and 75, 37ms for the hybrid, incremental and proposed methods, respectively. The CGH computation time per frame has thus been reduced by 48, 9% and 59, 9% using the incremental and proposed methods, respectively. This confirms that the proposed method is faster than the incremental method for the second part of the video.

Finally, for the third part of the video, the average number of calculated scene points per frame is equal to 215296 for all the compared methods. As a consequence, the average CGH computation time per frame is equal to 164, 57ms, 492, 80ms and 168, 07ms for the hybrid, incremental and proposed methods, respectively. The proposed method is thus much more efficient than the incremental method when there exist important changes between consecutive frames of the input 3D video. The CGH computation time per frame has been increased by 2% using the proposed method compared to the hybrid method. This is due to the additional pre-processing stage used to track the differences from one frame to another.

These experimental results show that for the 3D video cases in which consecutive frames differ slightly from each other, the number of scene points to be calculated and the CGH computation time can be significantly reduced using the proposed method. Additionally, in the other cases, the proposed method is still more efficient than the methods based on point-source or wave-field approaches.

### 5.3 Conclusion

In this chapter, we developed our second contribution, which combines temporal redundancy removal and hybrid point-source/wave-field approaches to further reduce the calculation time of video holograms. Our algorithm consists of three steps. First, intensity and depth data of the current 3D video frame are extracted and compared with those of the previous frame to remove temporally redundant data. Then, the CGH pattern for this compressed frame is generated using the hybrid point-source/wave-field method. The resulting CGH pattern is finally transmitted to the video output and stored in the previous frame buffer.

Experimental results show that for the 3D video cases in which consecutive frames differ slightly from each other, the number of scene points to be processed can be significantly reduced using the proposed method. As a consequence, the average calculation time per frame has been reduced up to 16, 60ms for CGHs of  $(2048 \times 2048)$  pixels. In other words, the proposed method is able to produce colorful CGHs at 60 frames per second, which is sufficient for real-time systems. Additionally, for the 3D video cases in which there exist large changes between consecutive frames, the proposed method is still more efficient than the methods based on point-source or wave-field approaches.

## **Part IV**

# **Hologram visual quality improvement**



## Chapter 6

# Hologram with occlusion effect using the hybrid approach

As stated in Chapter 4, the hybrid method outperforms both point-source and wave-field approaches without producing any visible artifact. Moreover, as shown in the previous chapter, when combined with temporal redundancy compression, this method is able to produce video CGHs in real-time. However, the original hybrid method does not take into account occlusions between objects in the scene.

In fact, in both Chapters 4 and 5, occlusions were naturally recorded in the input views and associated depth maps. Unfortunately, this hidden surface removal procedure is only valid for small viewing angles. Indeed, unlike still CG images, CGHs provide motion parallax, which is coupled with occlusion effect. Therefore, the occlusion effect in CGH implies that the visibility of objects changes according to the movement of the viewer. Since the computed holograms in Chapters 4 and 5 had pixel pitch  $p = 8, 1\mu\text{m}$ , their maximum viewing angle was limited to  $3.3^\circ$ , and no additional computation was needed to provide the occlusion effect. However, when computing CGHs with a smaller pixel pitch, the original hybrid method would produce occlusion holes and leakage fields for large viewing angles.

Accordingly, in this chapter we propose a CGH computation method with occlusion processing based on the hybrid point-source/wave-field approach. This chapter is organized as follows. Section 6.1 presents the proposed method, Section 6.2 gives a detailed description of the GPU implementation, and the experimental results are analyzed in Section 6.3.

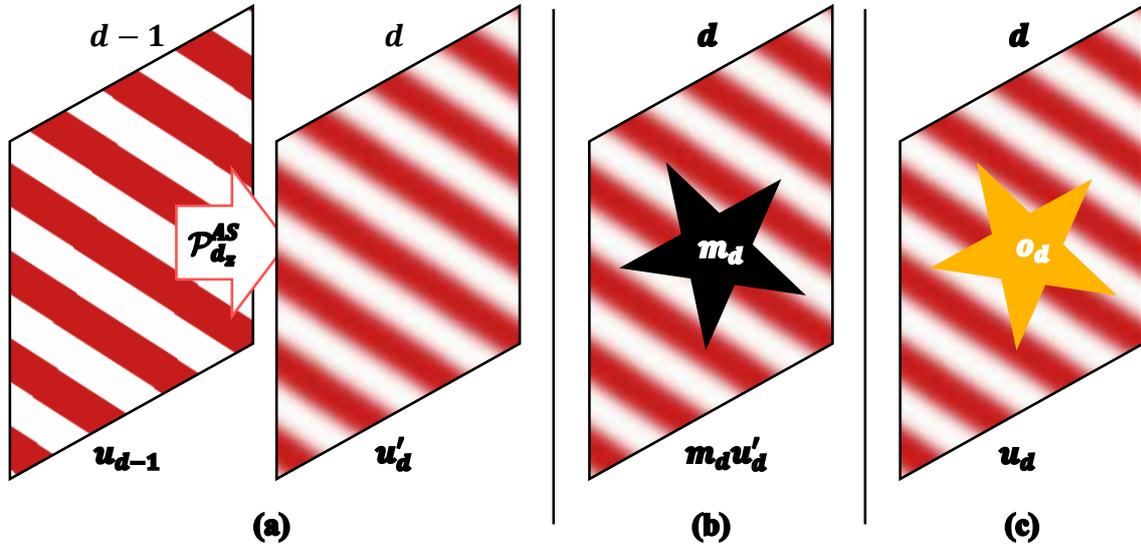


Fig. 6.1 Light shielding procedure. (a) Light scattered by layer  $d-1$  is numerically propagated towards layer  $d$ . (b) Light shielding is performed by multiplying the light field in layer  $d$  by a binary mask function  $m_d$  that has value 0 on the obstacle and 1 elsewhere. (c) The complex wave  $o_d$  scattered by layer  $d$  is added to the result.

## 6.1 Proposed method

The scene geometry and coordinate system used by the proposed method are the same as those used in the original hybrid point-source/wave-field method presented in Chapter 4. These are shown in Figure 4.1. In this section, we present how occlusion processing, also called light shielding, can be integrated during object wave computation using the hybrid approach. We develop our method from the simplest case, in which all the depth layers are wave-field layers, to the most general case, in which several point-source layers are stacked between each two wave-field layers.

### 6.1.1 Case 1: Light shielding by wave-field layers

We first consider the case in which all the depth layers contain a sufficient number of scene points to be considered as wave-field layers. In that case, light shielding is performed using a binary cross-section mask function, as proposed in [105].

The procedure is given in Figure 6.1. An occluding obstacle is located within layer  $d$ . The light wave  $u'_d$  incident on layer  $d$  is given by

$$u'_d(x, y) = \mathcal{P}_{d_z}^{AS} \{u_{d-1}\}(x, y), \quad (6.1)$$

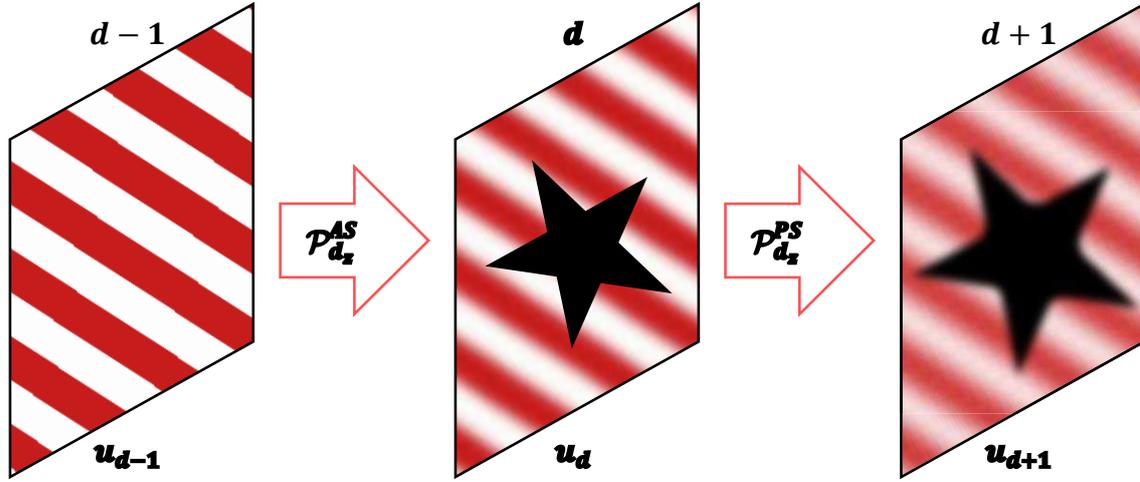


Fig. 6.2 Light shielding using the binary mask function. In this figure, we assume that  $o_d = 0$ .

where  $u_{d-1}$  is the total complex wave scattered by layer  $d-1$ .

Part of  $u'_d$  is shielded by the occluding scene points and its amplitude vanishes in the area of the obstacle. This is expressed by multiplying  $u'_d$  by a binary mask function  $m_d$  that has value 0 on the obstacle and 1 elsewhere. If the scene points within layer  $d$  also emit a wave  $o_d$ , the total complex wave  $u_d$  scattered by layer  $d$  is therefore given by

$$u_d(x, y) = o_d(x, y) + m_d(x, y)u'_d(x, y). \quad (6.2)$$

Light scattered by the scene is therefore numerically propagated and shielded from one layer to another using the recurrence formula

$$\begin{cases} u_0(x, y) = o_0(x, y) \\ u_d(x, y) = o_d(x, y) + m_d(x, y)\mathcal{P}_{d_z}^{AS} \{u_{d-1}\}(x, y) \quad \forall d > 0 \end{cases} \quad (6.3)$$

### 6.1.2 Case 2: One point-source layer between two wave-field layers

Let's now consider the case in which a single point-source layer  $d$  is located between two wave-field layers  $d-1$  and  $d+1$ . The procedure is given in Figure 6.2. An occluding obstacle is located within layer  $d$ . A straightforward way to compute light shielding by layer  $d$  is to use Eq. (6.2). The complex wave incident on layer  $d+1$  is therefore given by

$$\begin{aligned} u'_{d+1}(x, y) &= \mathcal{P}_{d_z}^{PS} \{u_d\}(x, y) \\ u'_{d+1}(x, y) &= \mathcal{P}_{d_z}^{PS} \{o_d + m_d \mathcal{P}_{d_z}^{AS} \{u_{d-1}\}\}(x, y). \end{aligned} \quad (6.4)$$

However, while being perfectly adapted to wave-field layers, whose complex wave is numerically propagated using  $\mathcal{P}_z^{AS}$ , light shielding using the binary mask function is not suited to point-source layers. Indeed, at each diffraction step, the complex wave scattered by the scene spreads gradually from one layer to another, and the wave  $u'_d$  incident on layer  $d$  may spread out on a large number of samples. Thus, even if layer  $d$  contains only a few scene points for which  $o_d$  is non-zero, the total complex wave  $u_d$  scattered by this layer may have a large number of non-zero values. Each non-zero value of  $u_d$  is therefore considered as a spherical light source by  $\mathcal{P}_z^{PS}$ . Since the computational complexity of  $\mathcal{P}_z^{PS}$  is dependent on the number of spherical light sources within the source layer, this light shielding technique is highly inefficient for point-source layers.

Instead, to compute light shielding by point-source layers efficiently, we use a binary aperture function  $a_d$  given by

$$a_d(x, y) = 1 - m_d(x, y). \quad (6.5)$$

The procedure for light shielding using the binary aperture function is given in Figure 6.3. By substituting Eq. (6.5) in Eq. (6.4), the complex wave  $u'_{d+1}$  incident on layer  $d+1$  becomes

$$\begin{aligned} u'_{d+1}(x, y) &= \mathcal{P}_{d_z}^{PS} \{o_d + (1 - a_d)\mathcal{P}_{d_z}^{AS} \{u_{d-1}\}\} (x, y) \\ &= \mathcal{P}_{d_z}^{PS} \{\mathcal{P}_{d_z}^{AS} \{u_{d-1}\} + o_d - a_d\mathcal{P}_{d_z}^{AS} \{u_{d-1}\}\} (x, y) \\ &= \mathcal{P}_{2d_z}^{AS} \{u_{d-1}\} (x, y) + \mathcal{P}_{d_z}^{PS} \{o_d - a_d\mathcal{P}_{d_z}^{AS} \{u_{d-1}\}\} (x, y) \end{aligned} \quad (6.6)$$

since

$$\mathcal{P}_{d_z}^{PS} \equiv \mathcal{P}_{d_z}^{AS} \quad \text{and} \quad \mathcal{P}_{d_z}^{PS} \{\mathcal{P}_{d_z}^{AS} \{u_{d-1}\}\} \equiv \mathcal{P}_{2d_z}^{AS} \{u_{d-1}\}, \quad (6.7)$$

according to definitions of  $\mathcal{P}_z^{AS}$  and  $\mathcal{P}_z^{PS}$ . Let us denote

$$\hat{u}_d(x, y) = o_d(x, y) - a_d(x, y)\mathcal{P}_{d_z}^{PS} \{u_{d-1}\} (x, y). \quad (6.8)$$

Eq. (6.6) thus becomes

$$u'_{d+1}(x, y) = \mathcal{P}_{2d_z}^{AS} \{u_{d-1}\} (x, y) + \mathcal{P}_{d_z}^{PS} \{\hat{u}_d\} (x, y). \quad (6.9)$$

The calculation of  $u'_{d+1}$  using the binary aperture involves computing one more propagation than using the binary mask. However,  $\mathcal{P}_{d_z}^{PS} \{u_{d-1}\}$  needs to be calculated only within the region defined by aperture  $a_d$ , which corresponds to the coordinates of scene points located within layer  $d$ . Therefore, the number of non-zero values of  $\hat{u}_d$  remains equal to the number of scene points for which  $o_d$  is non-zero. Using the binary aperture function has the great

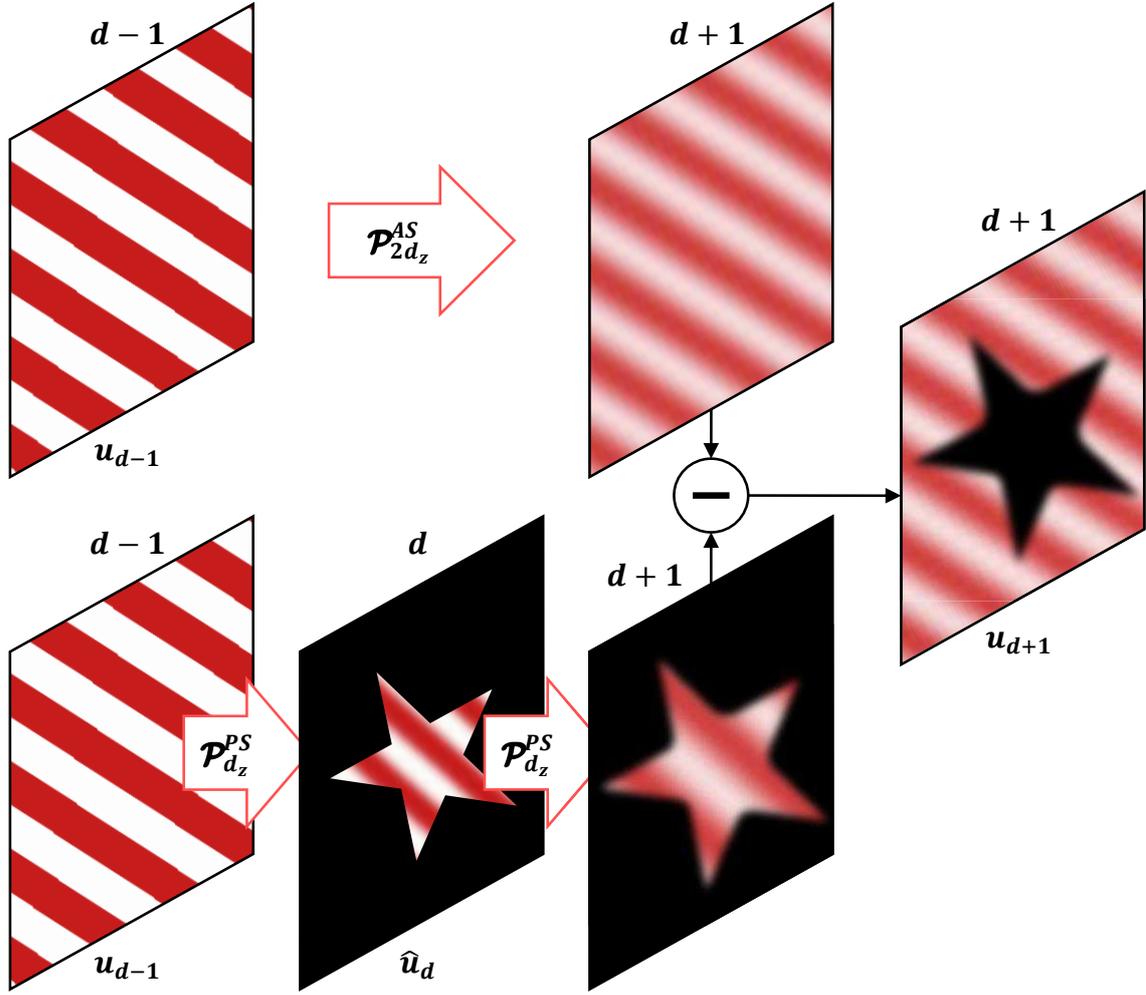


Fig. 6.3 Light shielding using the binary aperture function. In this figure, we assume that  $o_d = 0$ .

advantage of not increasing the number of spherical light sources for which the complex wave has to be computed at each diffraction step. This technique is therefore highly efficient for light shielding by point-source layers.

Light scattered by the scene is therefore numerically propagated and shielded from one layer to another, using the recurrence formula

$$\begin{cases} u_0 = o_0 \\ \hat{u}_d = o_d - a_d \mathcal{P}_{d_z}^{PS} \{u_{d-1}\} \\ u_d = o_d + m_d (\mathcal{P}_{2d_z}^{AS} \{u_{d-2}\} + \mathcal{P}_{d_z}^{PS} \{\hat{u}_{d-1}\}) \end{cases} \quad \forall d > 0, d \in S \quad (6.10)$$

where  $S$  and  $W$  are the sets of point-source and wave-field layers, respectively.

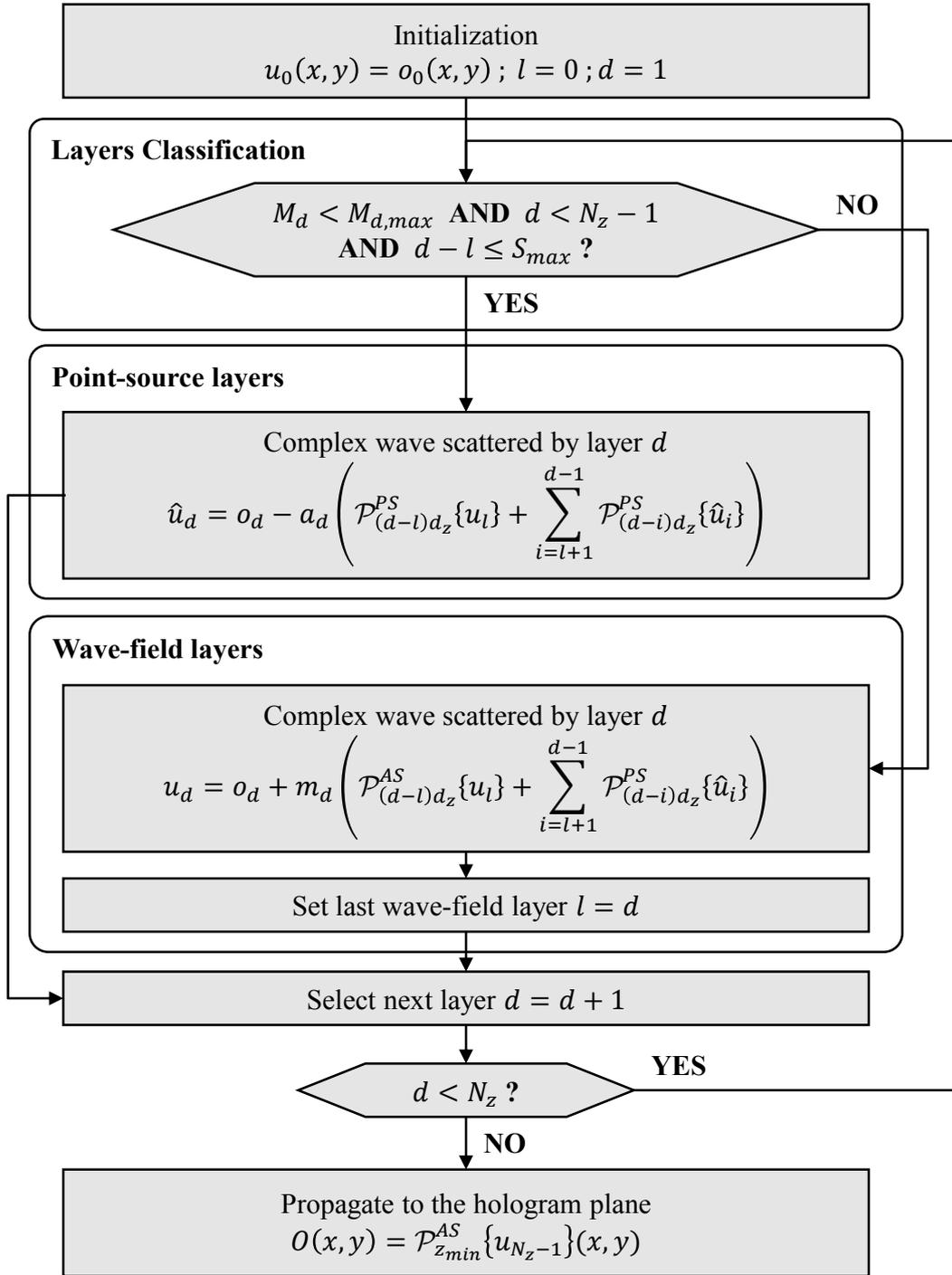


Fig. 6.4 Block-diagram of the proposed method.

### 6.1.3 Case 3: Several point-source layers between each two wave-field layers

We now consider the most general case, in which several point-source layers are stacked between each two wave-field layers. The overall block-diagram of the method is shown in Figure 6.4. Similarly to the original hybrid method, if the number of points  $M_d$  within layer  $d$  is smaller than the threshold value  $M_{d,\max}$ , this layer is considered to be a point-source layer, otherwise it is considered to be a wave-field layer.

To compute light scattered by the scene, Eq. (6.10) is generalized to

$$\begin{cases} u_0 = o_0 \\ \hat{u}_d = o_d - a_d \left( \mathcal{P}_{(d-l)d_z}^{PS} \{u_l\} + \sum_{i=l+1}^{d-1} \mathcal{P}_{(d-i)d_z}^{PS} \{\hat{u}_i\} \right) & \forall d, 0 \leq l < d, d \in S \\ u_d = o_d + m_d \left( \mathcal{P}_{(d-l)d_z}^{AS} \{u_l\} + \sum_{i=l+1}^{d-1} \mathcal{P}_{(d-i)d_z}^{PS} \{\hat{u}_i\} \right) & \forall d, 0 \leq l < d, d \in W \end{cases} \quad (6.11)$$

where  $l$  is the last wave-field layer. Since the binary aperture  $a_d$  has value 1 only at the coordinates of scene points located within layer  $d$ ,  $\hat{u}_d$  needs to be calculated within this region only. The computational complexity of this calculation is thus  $O(M_d M + M_d N_{T,(d-l)d_z})$ , where  $M$  is the total number of scene points within the depth layers located between  $d$  and the last wave-field layer  $l$ :  $M = \sum_{i=l+1}^{d-1} M_i$ .

Finally, the complex wave scattered by layer  $N_z - 1$  is numerically propagated to the hologram plane to obtain the object wave  $O$ , according to

$$O(x, y) = \mathcal{P}_{z_{\min}}^{AS} \{u_{N_z-1}\}(x, y), \quad (6.12)$$

and the CGH is computed using the bipolar intensity formula [107], such that

$$H(x, y) = \Re \{O(x, y)R^*(x, y)\}. \quad (6.13)$$

Let  $k = d - l - 1$  be the number of point-source layers between the last wave-field layer  $l$  and the current layer  $d$ . We call  $P(n)$  the property which states that Eq. (6.11) and Eq. (6.3) give the same complex field  $u_d$  for all  $k \leq n$ . We demonstrate that  $P(n)$  holds for all  $n \geq 0$ . For  $n = 0$ , Eq. (6.11) becomes

$$u_d = o_d + m_d \mathcal{P}_{d_z}^{AS} \{u_{d-1}\} \quad \forall d > 0, \quad (6.14)$$

which we recognize to be equal to  $u_d$  given by Eq. (6.3).  $P(n)$  therefore holds for  $n = 0$ .

We now assume that  $P(n)$  holds. For  $k = n + 1$ , we therefore have

$$\begin{aligned}
u_{d+1} &= o_{d+1} + m_{d+1} \mathcal{P}_{d_z}^{AS} \{u_d\} \\
&= o_{d+1} + m_{d+1} \mathcal{P}_{d_z}^{AS} \left\{ o_d + (1 - a_d) \left( \mathcal{P}_{(d-l)d_z}^{AS} \{u_l\} + \sum_{i=l+1}^{d-1} \mathcal{P}_{(d-i)d_z}^{PS} \{\hat{u}_i\} \right) \right\} \\
&= o_{d+1} + m_{d+1} \left( \mathcal{P}_{d_z}^{PS} \{\hat{u}_d\} + \mathcal{P}_{(d+1-l)d_z}^{AS} \{u_l\} + \sum_{i=l+1}^{d-1} \mathcal{P}_{(d+1-i)d_z}^{PS} \{\hat{u}_i\} \right) \\
&= o_{d+1} + m_{d+1} \left( \mathcal{P}_{(d+1-l)d_z}^{AS} \{u_l\} + \sum_{i=l+1}^d \mathcal{P}_{(d+1-i)d_z}^{PS} \{\hat{u}_i\} \right), \tag{6.15}
\end{aligned}$$

where

$$\begin{aligned}
\hat{u}_{d+1} &= o_{d+1} - a_{d+1} \mathcal{P}_{d_z}^{PS} \{u_d\} \\
&= o_{d+1} - a_{d+1} \mathcal{P}_{d_z}^{PS} \left\{ o_d + (1 - a_d) \left( \mathcal{P}_{(d-l)d_z}^{AS} \{u_l\} + \sum_{i=l+1}^{d-1} \mathcal{P}_{(d-i)d_z}^{PS} \{\hat{u}_i\} \right) \right\} \\
&= o_{d+1} - a_{d+1} \left( \mathcal{P}_{d_z}^{PS} \{\hat{u}_d\} + \mathcal{P}_{(d+1-l)d_z}^{PS} \{u_l\} + \sum_{i=l+1}^{d-1} \mathcal{P}_{(d+1-i)d_z}^{PS} \{\hat{u}_i\} \right) \\
&= o_{d+1} - a_{d+1} \left( \mathcal{P}_{(d+1-l)d_z}^{PS} \{u_l\} + \sum_{i=l+1}^d \mathcal{P}_{(d+1-i)d_z}^{PS} \{\hat{u}_i\} \right). \tag{6.16}
\end{aligned}$$

Eq. (6.15) and Eq. (6.16) show that if  $P(n)$  holds,  $P(n+1)$  also holds. As a consequence,  $P(n)$  holds for all  $n \geq 0$ . Since we use the same succession of layers as defined in the formulation of  $P(n)$ , our method gives the exact same results as those given by Eq. (6.3).

### 6.1.4 Layers classification

To implement the proposed method, it is necessary to determine the value of  $M_{d,\max}$ . We call  $t^{AS}$  the time needed to compute light propagation and light shielding for layer  $d$  using the wave-field approach, and  $t^{PS}$  the time needed to compute it using the point-source approach.

$t^{AS}$  is only dependent on the number of hologram pixels  $N = (N_x \times N_y)$  and is given by

$$t^{AS}(N) = \alpha N \log(N) + \beta N, \tag{6.17}$$

where  $\alpha$  and  $\beta$  are constant coefficients.

On the other hand,  $t^{PS}$  is given as the sum of the time needed to compute the propagation of complex wave scattered by layer  $d$  using  $\mathcal{P}^{PS}$  and the time needed to compute light shielding by layer  $d$  using aperture function  $a_d$ . It is therefore given by

$$t^{PS}(M_d, N_{T,d_z}) = \gamma (M_d N_{T,d_z} + M_d M + M_d N_{T,(d-l)d_z}), \quad (6.18)$$

where  $\gamma$  is a constant coefficient and  $M$  is the total number of scene points within the depth layers located between  $d$  and the last wave-field layer  $l$ . The values of  $\alpha$ ,  $\beta$  and  $\gamma$  are dependent on the implementation and on the computing system used.

To maximize the efficiency of the proposed method,  $M_{d,\max}$  must be set such that

$$t^{AS}(N) = t^{PS}(M_{d,\max}, N_{T,d_z}) \Leftrightarrow M_{d,\max} = \frac{\alpha N \log(N) + \beta N}{\gamma(N_{T,d_z} + M + N_{T,(d-l)d_z})}. \quad (6.19)$$

To find the numerical values of the coefficients  $\alpha$ ,  $\beta$  and  $\gamma$ , we measured the calculation time of the point-source and wave-field approaches for one million randomly generated depth layers with different numbers of scene points. We then found these values using the Gnuplot implementation of the nonlinear least-squares Levenberg-Marquardt algorithm [110].

It must be noted that the calculation of Eq. (6.11) requires the complex waves  $\hat{u}_i$  scattered by depth layers  $i \in \{l + 1, \dots, d - 1\}$  to be kept in memory. Since this may require a huge amount of memory, we restrict the number of consecutive point-source layers to be less or equal to a maximum value  $S_{\max}$ . The value of  $S_{\max}$  may be adjusted depending on the maximum memory amount of the computing system used.

## 6.2 Graphics Processing Unit implementation

Like the original hybrid method, the proposed method was implemented in C++/CUDA on a PC system employing an Intel Core i7-4930K CPU operating at 3.40 GHz, a main memory of 16 GB, three GPUs NVIDIA GeForce GTX 780Ti, and an operating system of Microsoft Windows 8.

To compute CGH patterns for the three colors simultaneously, we used one CPU thread and one GPU per color. In the implementation, all the calculation is done by the GPUs. The CPU threads are only used to load the input 3D scene, launch CUDA kernels and save the computed CGH into an output file. Finally, to achieve best performance on the GPU, all the computations are performed using single precision.

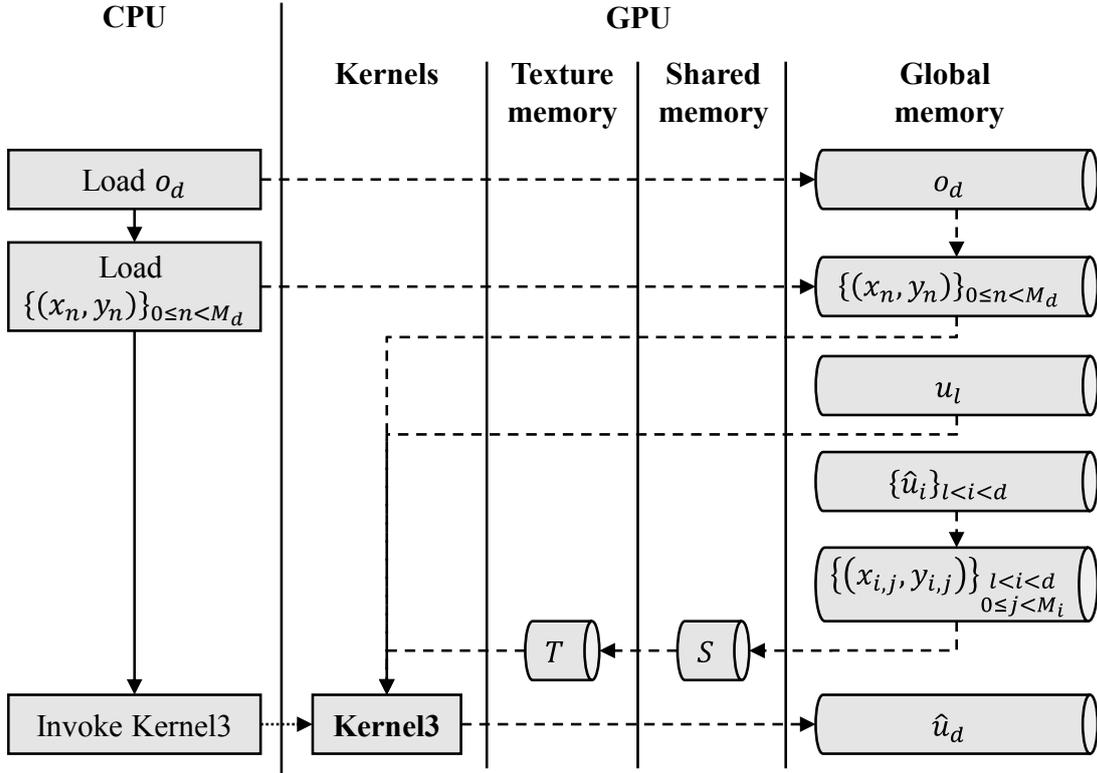


Fig. 6.5 Software structure for the calculation of  $\hat{u}_d$  when  $d$  is a point-source layer.

### 6.2.1 Point-source layers

Figure 6.5 shows the software structure for the calculation of the complex wave  $\hat{u}_d$  scattered by point-source layer  $d$ . First, the CPU loads the complex amplitude  $o_d$  and coordinates  $\{(x_n, y_n)\}_{0 \leq n < M_d}$  of scene points located within layer  $d$  into the GPU global memory. It must be noted that since the object wave is numerically propagated from one layer to another, the complex waves  $u_l$ ,  $\{\hat{u}_i\}_{l < i < d}$  and coordinates  $\{(x_{i,j}, y_{i,j})\}_{\substack{l < i < d \\ 0 \leq j < M_i}}$  of points located between  $d$  and the last wave-field layer  $l$  are already stored into the GPU global memory.

The CPU then invokes kernel *Kernel3*, whose pseudo-code is given in Algorithm 3. *Kernel3* computes light shielding for every scene point within layer  $d$  in parallel, with one thread per point. Since this computation involves one calculation per scene point within layer  $d$  per point located between  $d$  and  $l$ , all the threads executing *Kernel3* must access  $\{\hat{u}_i\}_{l < i < d}$  and  $\{(x_{i,j}, y_{i,j})\}_{\substack{l < i < d \\ 0 \leq j < M_i}}$  data simultaneously. Therefore, to optimize the data transfers, *Kernel3* starts out by loading  $\{(x_{i,j}, y_{i,j}, \hat{u}_i(x_{i,j}, y_{i,j}))\}_{\substack{l < i < d \\ 0 \leq j < M_i}}$  into the shared memory. Once this is done, threads must be synchronized to wait until all the data has been properly loaded into shared memory. Then, *Kernel3* fetches  $T$  to compute light shielding.

**Algorithm 3** *Kernel3* pseudo-code

---

**Require:**  $l, d, d_z$   
**Require:**  $o_d, u_l, \{\hat{u}_i\}_{l < i < d}$   
**Require:**  $\{(x_n, y_n)\}_{0 \leq n < M_d}$   
**Require:**  $\{(x_{i,j}, y_{i,j})\}_{\substack{l < i < d \\ 0 \leq j < M_i}}$   
**Ensure:**  $\hat{u}_d$

```

1: for  $i \in \{l + 1, \dots, d - 1\}$  in parallel do                                ▷ One thread per layer
2:   for  $j \in \{0, \dots, M_i - 1\}$  do                                       ▷ For each scene point within the layer
3:      $x_{i,j}^s \leftarrow x_{i,j}$                                                ▷ Load  $x_{i,j}$  in shared memory
4:      $y_{i,j}^s \leftarrow y_{i,j}$                                                ▷ Load  $y_{i,j}$  in shared memory
5:      $\hat{u}_{i,j}^s \leftarrow \hat{u}_i(x_{i,j}, y_{i,j})$                                ▷ Load  $\hat{u}_i(x_{i,j}, y_{i,j})$  in shared memory
6:   end for
7: end for
8: Synchronize threads
9: for  $n \in \{0, \dots, M_d - 1\}$  in parallel do                                ▷ One thread per scene point
10:   $r \leftarrow o_d(x_n, y_n)$                                                ▷ Load  $o_d$  in register
11:   $Z \leftarrow (d - l)d_z$ 
12:  for  $X \in [-R_{\max}(Z); R_{\max}(Z)]$  do
13:    for  $Y \in [-R_{\max}(Z); R_{\max}(Z)]$  do
14:       $x \leftarrow x_n + X$ 
15:       $y \leftarrow y_n + Y$ 
16:       $r \leftarrow r - u_l(x, y)T(|X|, |Y|, Z)$                                ▷ Fetch the LUT in texture memory
17:    end for
18:  end for
19:  for  $i \in \{l + 1, \dots, d - 1\}$  do                                       ▷ For each layer
20:     $Z \leftarrow (d - i)d_z$ 
21:    for  $j \in \{0, \dots, M_i - 1\}$  do                                       ▷ For each scene point within the layer
22:       $X \leftarrow |x_n - x_{i,j}^s|$ 
23:       $Y \leftarrow |y_n - y_{i,j}^s|$ 
24:      if  $X < R_{\max}(Z)$  AND  $Y < R_{\max}(Z)$  then
25:         $r \leftarrow r - \hat{u}_{i,j}^s T(X, Y, Z)$                                ▷ Fetch the LUT in texture memory
26:      end if
27:    end for
28:  end for
29:   $\hat{u}_d(x_n, y_n) \leftarrow r$                                                ▷ Store  $\hat{u}_d$  in global memory
30: end for

```

---

As explained in Chapter 4,  $T$  is stored in the texture memory to achieve better performance. However, this time, since the number of consecutive point-source layers is limited by  $S_{\max}$ , the total amount of memory needed to store  $T$  is given by

$$N_T = 8S_{\max}N_{T, S_{\max}d_z}. \quad (6.20)$$

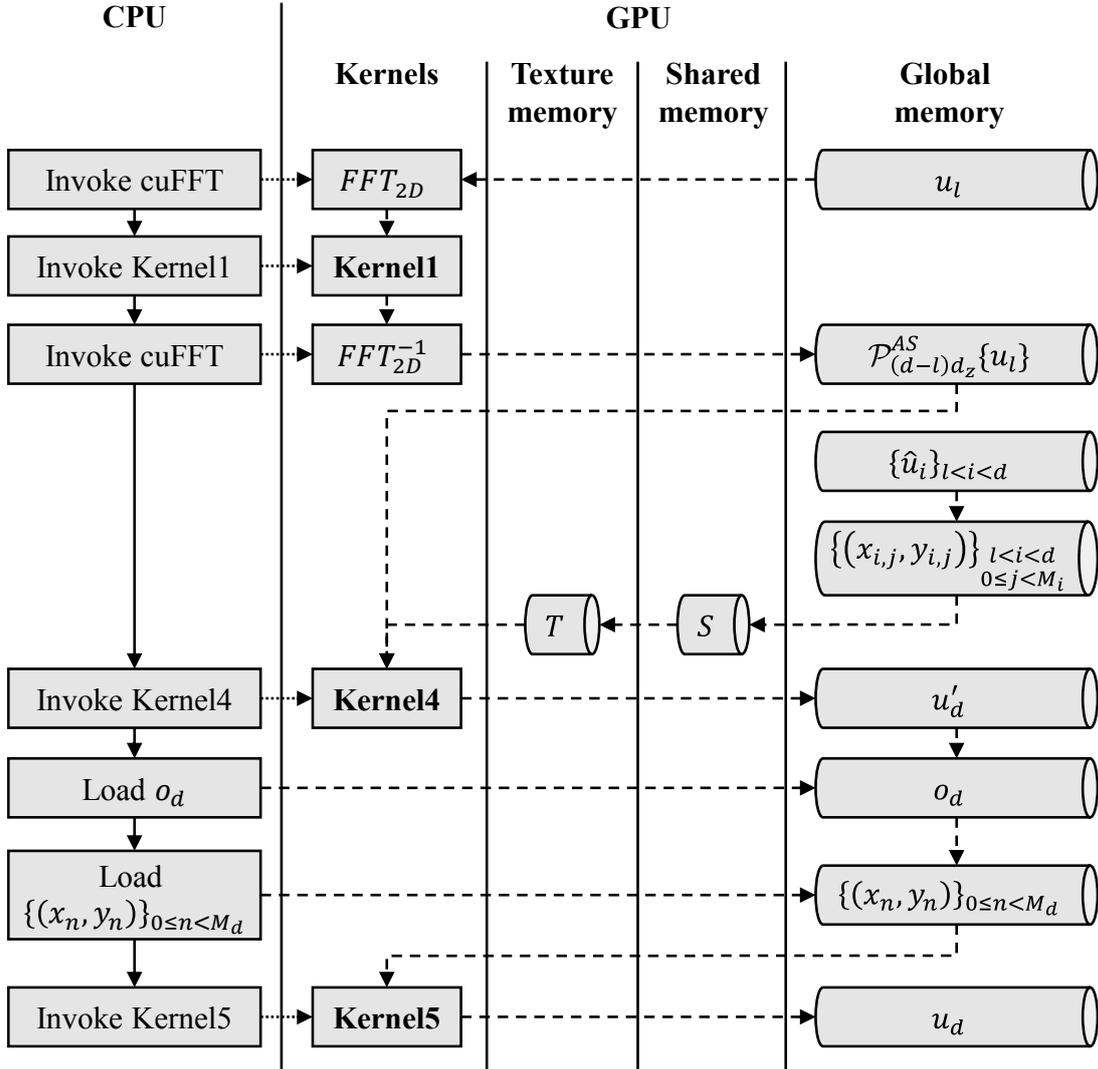


Fig. 6.6 Software structure for the calculation of  $u_d$  when  $d$  is a wave-field layer.

For  $S_{\max} = 100$ ,  $N_z = 1024$ ,  $\lambda = 640\text{nm}$ ,  $p = 4.8\mu\text{m}$  and  $z_{\max} = 5\text{cm}$ , the total amount of memory needed to store  $T$  is  $N_T = 3.7\text{MB}$ . The memory size of the LUT has been reduced by two orders of magnitude compared to the original hybrid method, which leaves room to keep in memory the complex waves  $u_l$  and  $\hat{u}_i$  scattered by layers  $i \in \{l + 1, \dots, d - 1\}$ .

*Kernel3* thus fetches the LUT and performs a multiply-accumulate operation using an on-chip register to limit the number of accesses to global memory. Finally, the result  $\hat{u}_d$  is stored into the global memory.

**Algorithm 4** *Kernel4* pseudo-code

---

**Require:**  $l, d, d_z$   
**Require:**  $\mathcal{P}_{(d-l)d_z}^{AS} \{u_l\}$   
**Require:**  $\{\hat{u}_i\}_{l < i < d}$   
**Require:**  $\{(x_{i,j}, y_{i,j})\}_{\substack{l < i < d \\ 0 \leq j < M_i}}$   
**Ensure:**  $u'_d$

- 1: **for**  $i \in \{l + 1, \dots, d - 1\}$  **in parallel do** ▷ One thread per layer
- 2:     **for**  $j \in \{0, \dots, M_i - 1\}$  **do** ▷ For each scene point within the layer
- 3:          $x_{i,j}^s \leftarrow x_{i,j}$  ▷ Load  $x_{i,j}$  in shared memory
- 4:          $y_{i,j}^s \leftarrow y_{i,j}$  ▷ Load  $y_{i,j}$  in shared memory
- 5:          $\hat{u}_{i,j}^s \leftarrow \hat{u}_i(x_{i,j}, y_{i,j})$  ▷ Load  $\hat{u}_i(x_{i,j}, y_{i,j})$  in shared memory
- 6:     **end for**
- 7: **end for**
- 8: Synchronize threads
- 9: **for all**  $(x, y)$  **in parallel do** ▷ One thread per object wave sample
- 10:      $r \leftarrow \mathcal{P}_{(d-l)d_z}^{AS} \{u_l\}(x, y)$  ▷ Load  $\mathcal{P}_{(d-l)d_z}^{AS} \{u_l\}$  in register
- 11:     **for**  $i \in \{l + 1, \dots, d - 1\}$  **do** ▷ For each layer
- 12:          $Z \leftarrow (d - i)d_z$
- 13:         **for**  $j \in \{0, \dots, M_i - 1\}$  **do** ▷ For each scene point
- 14:              $X \leftarrow |x - x_{i,j}^s|$
- 15:              $Y \leftarrow |y - y_{i,j}^s|$
- 16:             **if**  $X < R_{\max}(Z)$  **AND**  $Y < R_{\max}(Z)$  **then**
- 17:                  $r \leftarrow r + \hat{u}_{i,j}^s T(X, Y, Z)$  ▷ Fetch the LUT in texture memory
- 18:             **end if**
- 19:         **end for**
- 20:     **end for**
- 21:      $u'_d(x, y) \leftarrow r$  ▷ Store  $u_d$  in global memory
- 22: **end for**

---

**Algorithm 5** *Kernel5* pseudo-code

---

**Require:**  $u'_d$   
**Require:**  $o_d$   
**Require:**  $\{(x_n, y_n)\}_{0 \leq n < M_d}$   
**Ensure:**  $u_d$

- 1: **for all**  $(x, y)$  **in parallel do** ▷ One thread per object wave sample
- 2:      $r \leftarrow u'_d(x, y)$  ▷ Store  $u'_d$  in register
- 3:     **if**  $(x, y) \in \{(x_n, y_n)\}_{0 \leq n < M_d}$  **then**
- 4:          $r \leftarrow o_d(x, y)$  ▷ Occlusion processing
- 5:     **end if**
- 6:      $u_d(x, y) \leftarrow r$  ▷ Store  $u_d$  in global memory
- 7: **end for**

---

### 6.2.2 Wave-field layers

Figure 6.6 shows the software structure for the calculation of the total complex wave  $u_d$  scattered by wave-field layer  $d$ . First, the complex wave scattered by last wave-field layer  $l$  is propagated towards layer  $d$  using the CUDA cuFFT library and *Kernel1*, whose pseudo-code is given in Algorithm 1. Then, the CPU invokes kernel *Kernel4*, whose pseudo-code is given in Algorithm 4.

*Kernel4* computes the propagation of light scattered by point-source layers towards layer  $d$ . Like *Kernel2* and *Kernel3*, the complex amplitudes and coordinates of point-sources  $\{(x_{i,j}, y_{i,j}, \hat{u}_i(x_{i,j}, y_{i,j}))\}_{\substack{l < i < d \\ 0 \leq j < M_i}}$  are first loaded into the shared memory. Then, *Kernel4* fetches the LUT and performs a multiply-accumulate operation for each sample of  $u'_d$  in parallel. Finally, the result is summed with the complex wave  $\mathcal{P}_{(d-l)d_z}^{AS} \{u_l\}$  scattered by layer  $l$ . To limit the number of accesses to global memory, these operations are performed using an on-chip register, and the final result is stored only once into the global memory.

The CPU then loads the complex amplitude  $o_d$  and coordinates  $\{(x_n, y_n)\}_{0 \leq n < M_d}$  of scene points located within layer  $d$  into the GPU global memory and invokes *Kernel5*, whose pseudo-code is given in Algorithm 5. *Kernel5* computes light shielding by layer  $d$  using the binary mask function. For each sample of  $u'_d$  in parallel, if this sample is located within the region for which  $m_d(x, y) = 0$ , its amplitude is replaced by the point's amplitude  $o_d(x, y)$ .

## 6.3 Experimental results and discussion

Table 6.1 shows the hologram parameters used for the experiments. The hologram is sampled on a regular 2D grid of resolution  $(7680 \times 4320)$  with pixel pitch  $p = 4.8 \mu\text{m}$ . The wavelengths are set to 640nm, 532nm and 473nm for the Red, Green and Blue channels, respectively. Finally,  $S_{\max}$  is set to 100.

Table 6.1 Hologram synthesis parameters used for the experiments.

Scene parameters		Hologram parameters	
Location of the scene	(0, 4cm)	Hologram resolution	$(7680 \times 4320)$
Number of depth layers ( <i>Dice1</i> )	$N_z = 512$	Pixel pitch	$p = 4.8 \mu\text{m}$
Number of depth layers ( <i>Dice2</i> )	$N_z = 1024$	Red wavelength	640nm
Number of depth layers ( <i>City</i> )	$N_z = 2048$	Green wavelength	532nm
		Blue wavelength	473nm



Fig. 6.7 Computer graphics images of the three test scenes used for the experiments.

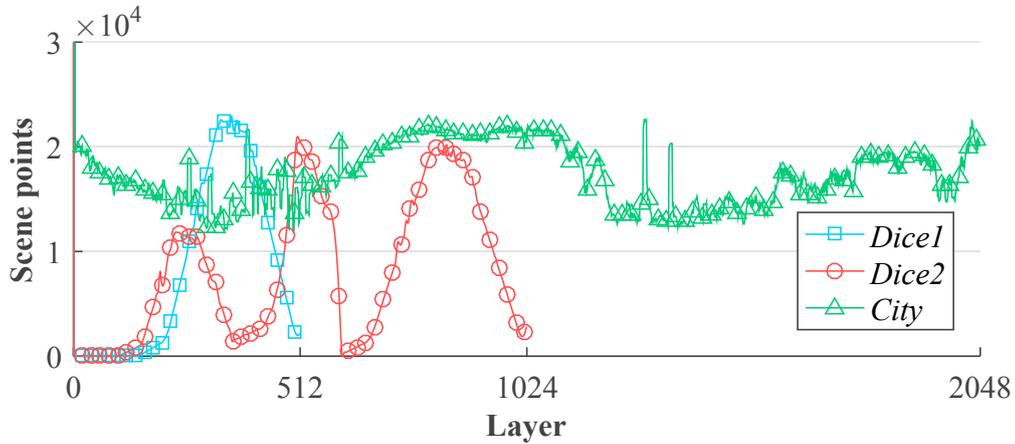


Fig. 6.8 Number of scene points per layer for the three scenes.

### 6.3.1 Input 3D scenes

For the experiments, we used three different test scenes, whose CG images are shown in Figure 6.7. The scenes are sliced into a set of  $N_z$  depth layers parallel to the hologram plane and located between  $z_{\min} = 0$  and  $z_{\max} = 4\text{cm}$ .  $N_z$  is set to 512, 1024 and 2048 for *Dice1*, *Dice2* and *City*, respectively. Figure 6.8 shows the number of scene points per layer for the three input scenes. As shown in this figure, all the depth layers contain scene points, except those located between the background and the first dice in *Dice1* and *Dice2*.

In the following, we compare the proposed method with GPU implementations of two other methods: (1) the point-source method without occlusion effect proposed in [81], which computes complex wave scattered by each layer using a point-source approach, and (2) the wave-field method with occlusion effect proposed in [105], which computes complex wave scattered by each layer using a wave-field approach. We adapted both methods to produce colorful CGH.

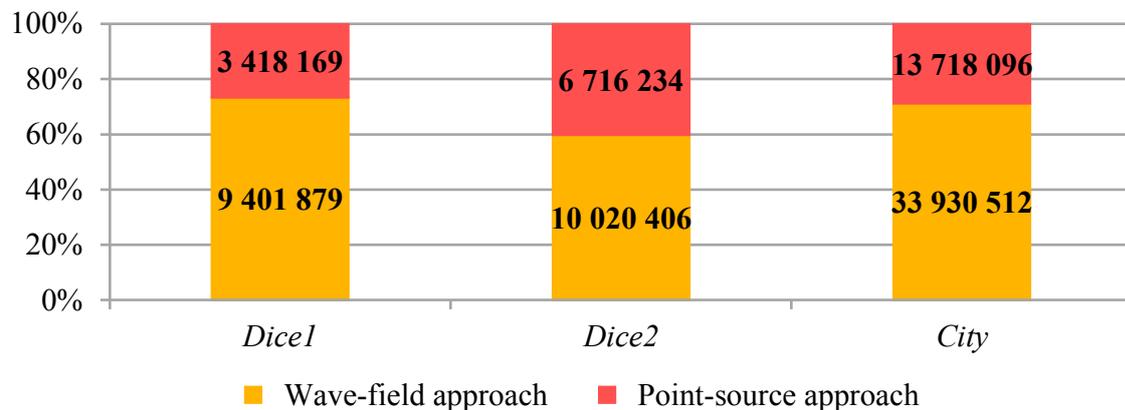


Fig. 6.9 Number and percentage of scene points processed by the proposed method using the point-source and wave-field approaches.

### 6.3.2 Analysis of the visual quality of the reconstructed images

Figure 6.9 shows the number and percentage of scene points processed by the proposed method using the point-source and wave-field approaches. As shown in this figure, the complex waves scattered by 73.3%, 59.9% and 71.2% of scene points were computed by the proposed method using the wave-field approach for *Dice1*, *Dice2* and *City*, respectively.

Figure 6.10 shows the scene images numerically reconstructed from the CGH patterns of the three test scenes generated by the point-source method (first row), the wave-field method (second row), and the proposed method (third row). As shown in the first row, the point-source method does not take into account occlusions in the scene, and the objects thus appear semi-transparent, strongly limiting the realism of the displayed image. On the other hand, occlusions between objects in the scene are accurately reproduced using the wave-field and proposed methods without producing any visible artifact (rows 2 and 3). These results confirm the need for CGH methods to properly take into account occlusions between objects in a scene to provide a realistic three-dimensional illusion.

To experimentally verify these numerical results, we displayed the generated hologram of scene *Dice2* on the NICT's 8K4K holographic display [208]. Figure 6.11 shows the optically reconstructed scene images recorded with a camera focusing on different depths. When the camera is focused on a given object, the rest of the scene is blurred, and vice versa. This demonstrates that the proposed method provides an accurate accommodation cue of the 3D scene. Figure 6.12 shows the optical reconstructions captured from three different viewpoints. As shown in this figure, the proposed method is able to produce motion parallax with correct occlusion effect.

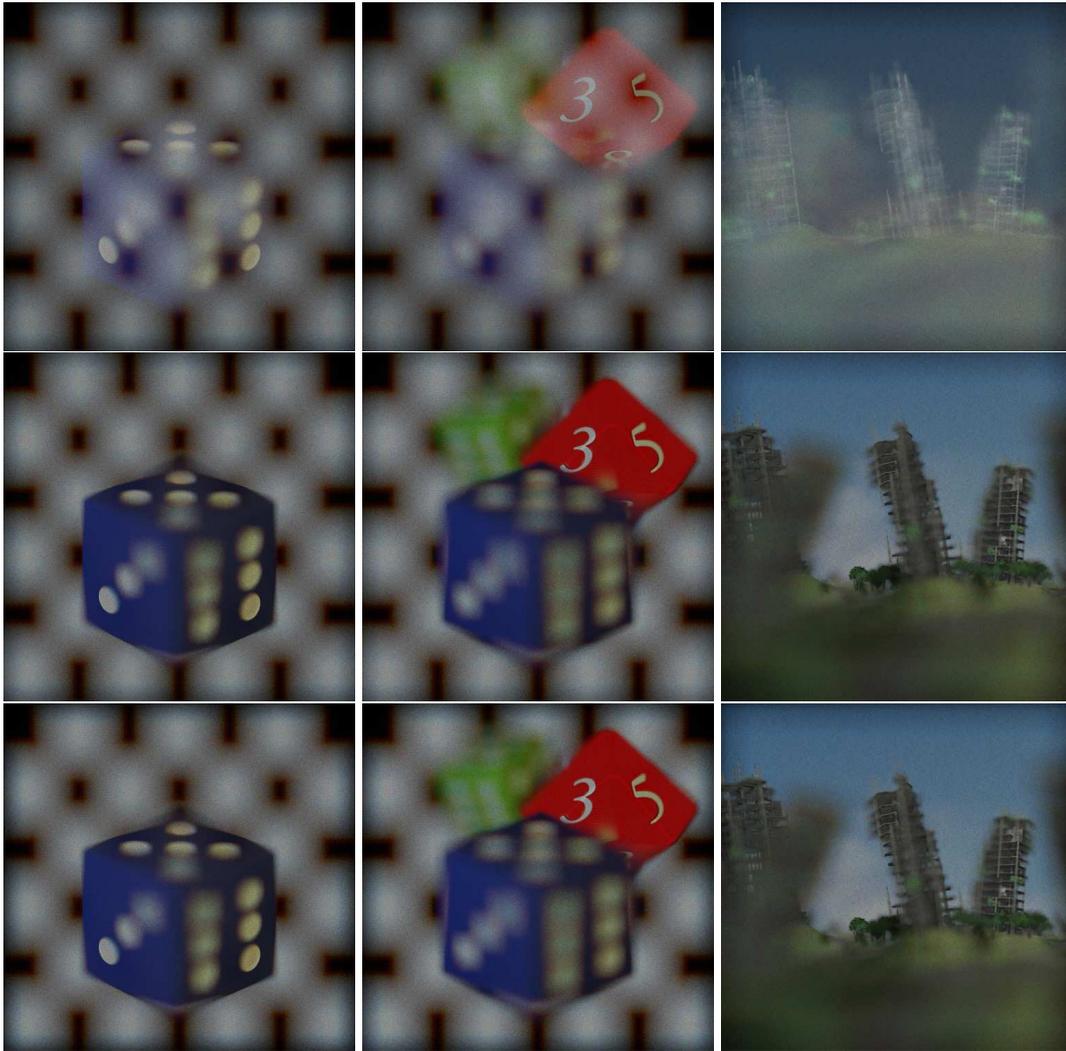


Fig. 6.10 Numerical reconstructions of the CGHs generated by the point-source method (first row), the wave-field method (second row), and the proposed method (third row).

### 6.3.3 Analysis of the hologram calculation time

Table 6.2 shows the CGH computation times for the three test scenes using the wave-field, point-source, and proposed methods. *Dice1* is sliced into 512 depth layers and is sampled by a total number of 12, 820, 048 points, *Dice2* is sliced into 1024 depth layers and is sampled by 16, 736, 640 points, and *City* is sliced into 2048 depth layers and is sampled by 47, 648, 608 points.

As shown in Table 6.2, using the wave-field method, the average CGH calculation time per layer remains constant for the three test scenes. This means that the total calculation time increases linearly with the number of layers and is not dependent on the total number of scene points. On the contrary, using the point-source method, the average CGH calculation



(a) Focused on the blue dice (b) Focused on the red dice (c) Focused on the green dice

Fig. 6.11 Optical reconstructions of scene *Dice2* focused on different objects. These pictures were taken with the NICT's 8K4K holographic display.



(a) Left view (b) Center view (c) Right view

Fig. 6.12 Optical reconstructions of scene *Dice2* captured from different viewpoints. These pictures were taken with the NICT's 8K4K holographic display.

time for one scene point remains constant for the three test scenes. This means that the total calculation time increases linearly with the number of scene points and is not dependent on the number of depth layers.

As a result, the point-source method is less efficient than the wave-field method when the number of points per layer is high, as in scenes *Dice1* and *City*, and is more efficient when the average number of points within each layer is low, as in scene *Dice2*. Using the point-source method, the total CGH calculation time has been increased by 8% and 2% for *Dice1* and *City*, and decreased by 27.4% for *Dice2*, compared to the wave-field method. It must be noted that the point-source approach that we implemented here does not take into account occlusions between objects in the scene. A light shielding operation, performed as a visibility test to check the existence of obstacles between an object point and each sampling point in the hologram plane, would have for effect to increase by a constant amount the average CGH calculation time per scene point.

Table 6.2 Comparison of the hologram computation times.

Scene	Number of scene points	Method	Calculation time		
			Total	Per layer	Per point
<i>Dice1</i>	12,820,048	Wave-field	21.79s (100%)	42.56ms	1.70 $\mu$ s
		Point-source	23.45s (108%)	45.80ms	1.83 $\mu$ s
		Proposed	12.22s (56.1%)	23.87ms	0.95 $\mu$ s
<i>Dice2</i>	16,736,640	Wave-field	42.22s (100%)	41.23ms	2.52 $\mu$ s
		Point-source	30.64s (72.6%)	29.92ms	1.83 $\mu$ s
		Proposed	21.79s (51.6%)	21.28ms	1.30 $\mu$ s
<i>City</i>	47,648,608	Wave-field	85.34s (100%)	41.67ms	1.79 $\mu$ s
		Point-source	86.99s (102%)	42.48ms	1.83 $\mu$ s
		Proposed	69.76s (81.7%)	34.06ms	1.46 $\mu$ s

As shown in Table 6.2, the total calculation time using the proposed method is dependent both on the number of layers and on the number of scene points. By combining these two approaches, the proposed method takes advantages from both of them and is therefore always more efficient. Using the proposed method, the total CGH calculation time has been decreased by 43.9%, 48.4% and 18.3% for *Dice1*, *Dice2* and *City*, respectively, compared to the wave-field method, and by 47.9%, 28.9% and 19.8% for *Dice1*, *Dice2* and *City*, respectively, compared to the point-source method. These experimental results confirm the performance superiority of the proposed method over the point-source and wave-field methods in terms of computation time.

## 6.4 Conclusion

In this chapter, we developed our third contribution, which aims at improving the visual quality of the displayed scene by taking into account occlusions between scene objects. Our algorithm consists of three steps. First, the 3D scene is sliced into several depth layers parallel to the hologram plane. Then, light scattered by the scene is propagated and shielded from one layer to another, starting from the farthest layer. For each depth layer, light propagation and light shielding are performed using either a point-source or a wave-field approach according to a threshold criterion on the number of points within the layer. Finally, we compute light propagation from the nearest layer to the hologram plane to obtain the CGH.

Numerical and optical reconstructions reveal that the proposed method accurately takes into account occlusions between objects in the scene and is able to produce a correct accommodation cue. Furthermore, the CGH calculation time has been reduced up to 47.9% and 48.4% compared to the point-source and wave-field approaches, respectively. This confirms the performance superiority of the proposed method over the point-source and wave-field approaches in terms of computation time.

# Chapter 7

## Hologram synthesis from MVD data with specular reflections

As stated in Chapter 3, it is still very challenging to compute CGHs of real existing scenes which provide all the human depth cues and accurate shading of the scene with only a few projection images.

Indeed, the point-source and wave-field approaches are able to produce CGHs with all the human depth cues and occlusion effect. However, to take into account shading of the scene, they require the 3D scene to be accurately modeled, limiting their application to synthetic scenes. On the other hand, since occlusions and shading of the scene are naturally recorded in the view projections, there is no need to perform any additional calculation to include them in the computed hologram using the multiple-viewpoint-projection and holographic stereogram approaches. However, since each projection image yields a single hologram pixel in the multiple-viewpoint-projection approach, a huge number of images are required, especially for high-resolution CGHs. Holographic stereograms require fewer projection images, but in this case the accommodation cue is sacrificed, which strongly limits the naturalism of the perceived 3D effect.

To overcome these limitations, we propose a novel approach for CGH computation with accurate shading of the scene from MVD data. This chapter is organized as follows. A preliminary version of the proposed method which can only render diffuse scenes is presented in Section 7.1, and the generalization of this method to scenes with specular reflections is described in Section 7.2. Finally, experimental results are analyzed in Section 7.3.

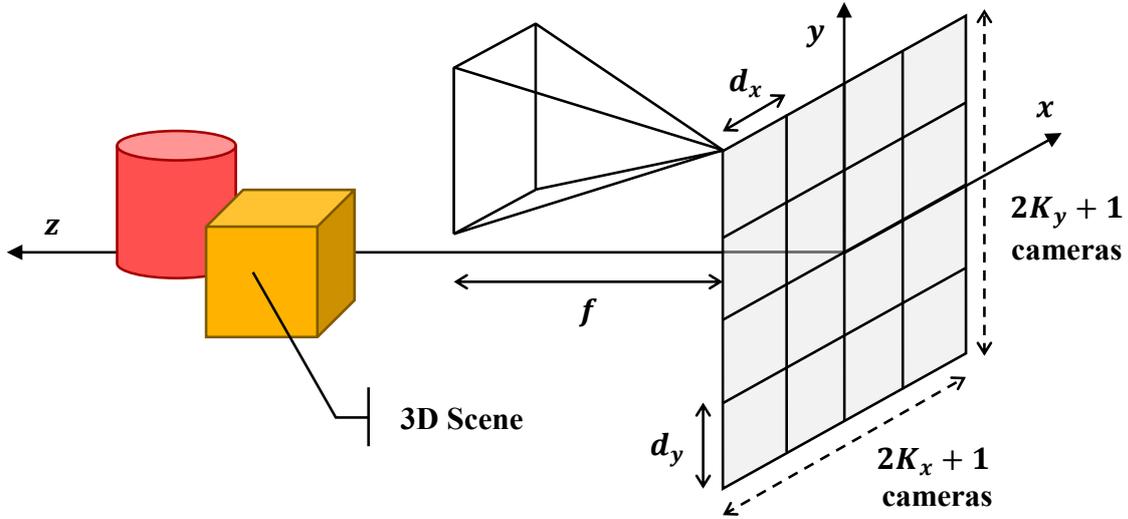


Fig. 7.1 Multiview-plus-Depth capture geometry.

## 7.1 Hologram synthesis for a diffuse scene

In this section, we limit ourselves to the rendering of diffuse scenes. The proposed method consists of three steps, which are detailed in the following. First, 2D-plus-depth projections of the scene are taken from different perspective viewpoints. Then, the 3D scene geometry is reconstructed from MVD data as a layered point-cloud. This 3D scene reconstruction step allows us to use only a few perspective projections of the scene without sacrificing any depth cue. Finally, light scattered by the scene is numerically propagated towards the hologram plane and encoded into the CGH.

### 7.1.1 Step 1: Multiview-plus-Depth capture

The first step of the proposed method is to capture 2D-plus-depth projections of the scene from different perspective viewpoints. Although other camera layouts may be used to acquire the MVD data, the proposed method uses  $(2K_x + 1) \times (2K_y + 1)$  cameras aligned in parallel on a 2D grid, as shown in Figure 7.1.

The coordinate system is defined by  $(x, y, z)$  so that the camera matrix lies on the  $(x, y, 0)$  plane and the 3D scene is located between  $z_{\min}$  and  $z_{\max}$ . We call  $d_x$  and  $d_y$  the distance between the cameras along the  $x$  and  $y$  axis, respectively, and  $f$  the focal length of the cameras. The acquired intensity views and depth maps have resolution  $(M_x \times M_y)$  and pixel pitch  $q$ . Finally, the cameras are numbered by  $(i, j)$ , with  $i \in \{-K_x, \dots, K_x\}$  and  $j \in \{-K_y, \dots, K_y\}$ .

The projection matrix of camera  $(i, j)$  is therefore given by

$$P_{i,j} = \begin{bmatrix} \alpha & 0 & u_0 & \alpha x_i \\ 0 & \alpha & v_0 & \alpha y_j \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (7.1)$$

where  $\alpha = f/q$  is the focal length expressed in pixels,  $(x_i, y_j)$  is the camera location, given by

$$x_i = id_x \quad \text{and} \quad y_j = jd_y, \quad (7.2)$$

and  $(u_0, v_0)$  corresponds to the principal point pixel's coordinates, given by

$$u_0 = \frac{M_x}{2} \quad \text{and} \quad v_0 = \frac{M_y}{2}. \quad (7.3)$$

### 7.1.2 Step 2: Scene geometry reconstruction

The second step of the proposed method is to reconstruct the 3D scene geometry from the MVD data. We call  $I_{i,j}$  and  $D_{i,j}$  the intensity and depth images captured by camera  $(i, j)$ . Since  $D_{i,j}$  is encoded as an 8-bits gray level image, each pixel  $(u, v)$  can be projected back to a 3D point whose coordinates are given by

$$\begin{pmatrix} x_{i,u} \\ y_{j,v} \\ z_d \end{pmatrix} = \begin{pmatrix} \frac{z(u-u_0)}{\alpha} - x_i \\ \frac{z(v-v_0)}{\alpha} - y_j \\ \frac{255-d}{255}(z_{\max} - z_{\min}) + z_{\min} \end{pmatrix}, \quad (7.4)$$

where  $d = D_{i,j}(u, v)$ .

The projected 3D points are thus naturally sliced into  $N_z = 256$  depth layers regularly located between  $z_{\min}$  and  $z_{\max}$  and separated by distance  $d_z$ . Furthermore, to improve the visual quality, each projected 3D point is attributed a Gaussian distribution, as proposed in [176]. Each depth layer  $d$  is therefore considered to operate as a surface source of light which emits a complex wave  $o_d$  given by

$$o_d(x, y) = \sum_{\substack{i,j,u,v \\ D_{i,j}(u,v)=d}} \sqrt{I_{i,j}(u, v)} \exp[j\phi(x, y)] \exp\left[-\frac{(x - x_{i,u})^2 + (y - y_{j,v})^2}{2s^2}\right], \quad (7.5)$$

where  $s^2$  is the variance of the Gaussian distribution, which controls the surface of each 3D point, and  $\phi(x, y) \in [0; 2\pi[$  is the initial phase, set to a uniform random value to keep the diffuseness of the scene.

### 7.1.3 Step 3: Light propagation and hologram encoding

The third step of the proposed method is to compute the propagation of light scattered by the scene towards the hologram plane. The coordinate system is defined by  $(x, y, z)$  so that the hologram lies on the  $(x, y, 0)$  plane. As stated before, the 3D scene is sliced into  $N_z = 256$  depth layers regularly located between  $z_{\min}$  and  $z_{\max}$  and separated by distance  $d_z$ . The depth layers are numbered from 0 to  $N_z - 1$ , from the farthest to the nearest to the hologram plane. The depth layers and the hologram are sampled on a regular 2D grid of resolution  $(N_x \times N_y)$  with pixel pitch  $p$ .

Light scattered by the scene is numerically propagated and shielded from one layer to another using the recurrence formula

$$\begin{cases} u_0(x, y) = o_0(x, y) \\ u_d(x, y) = o_d(x, y) + m_d(x, y) \mathcal{P}_{d_z}^{AS} \{u_{d-1}\}(x, y) \quad \text{for } d > 0 \end{cases} \quad (7.6)$$

where  $u_d$  is the total complex wave scattered by layer  $d$  and  $m_d$  is a binary mask function that has value 0 on the obstacle and 1 elsewhere.

Finally, the complex wave scattered by layer  $N_z - 1$  is numerically propagated to the hologram plane to obtain the object wave  $O$ , according to

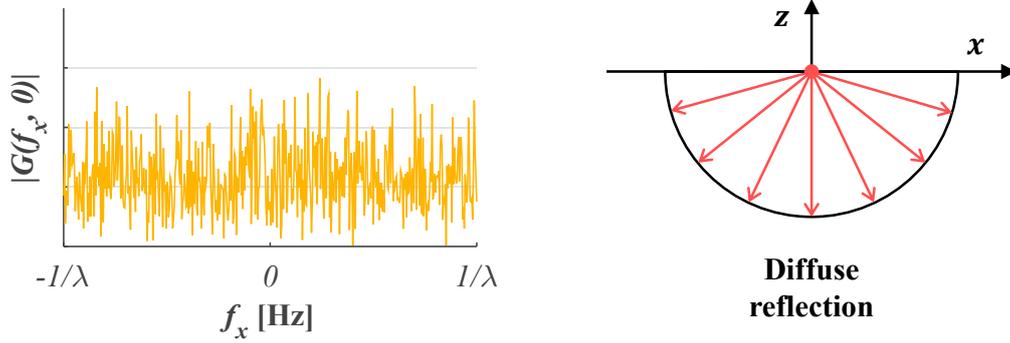
$$O(x, y) = \mathcal{P}_{z_{\min}}^{AS} \{u_{N_z-1}\}(x, y), \quad (7.7)$$

and the CGH is computed using the bipolar intensity formula [107], such that

$$H(x, y) = \Re \{O(x, y)R^*(x, y)\}. \quad (7.8)$$

## 7.2 Rendering specular reflections

This section describes the generalization of the proposed method to the rendering of specular reflections. Indeed, MVD data do not only capture the geometry of the scene but also its shading, including specular reflections. These variations of brightness are totally lost when using the diffuse-only version of the method presented in the previous section.



(a) Fourier spectrum of the original diffuser

(b) Illustration of diffuse reflection

Fig. 7.2 Relation between the spectrum of the original diffuser and diffuse reflection.

### 7.2.1 Role of the initial phase distribution

As stated before, each depth layer  $d$  is considered to operate as a surface source of light which emits a complex wave  $o_d$  given by Eq. (7.5). In this equation, the phase factor  $g(x, y) = \exp[j\phi(x, y)]$  plays the role of a diffuser that spreads light in space. As a consequence,  $\phi(x, y)$  is important for rendering specular reflections because it determines the diffuseness and direction of light emitted by each scene point. We call  $G$  the Fourier spectrum of  $g$ , given by

$$\begin{aligned} G(f_x, f_y) &= \mathcal{F}\{g\}(f_x, f_y) \\ &= \iint_{-\infty}^{\infty} g(x, y) \exp[-j2\pi(f_x x + f_y y)] dx dy, \end{aligned} \quad (7.9)$$

where  $f_x$  and  $f_y$  are the spacial frequencies with respect to the  $x$  and  $y$  axis, respectively.

The Angular Spectrum diffraction [53] expresses the propagation of light scattered by  $g$  over distance  $z$  as

$$\begin{aligned} g(x, y, z) &= \mathcal{F}^{-1}\{G(f_x, f_y) \exp(j2\pi f_z z)\}(x, y) \\ &= \iint_{-\infty}^{\infty} G(f_x, f_y) \exp[j2\pi(f_x x + f_y y + f_z z)] df_x df_y, \end{aligned} \quad (7.10)$$

where  $f_z = \sqrt{\lambda^{-2} - f_x^2 - f_y^2}$ .

To understand the behavior of  $g$ , we consider the time-independent complex form of a plane wave, given by

$$P(x, y, z) = A \exp(j\mathbf{k} \cdot \mathbf{r}), \quad (7.11)$$

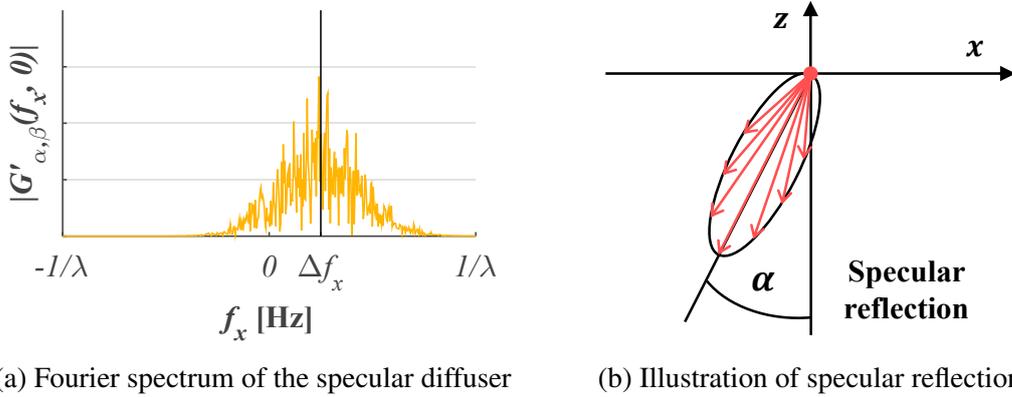


Fig. 7.3 Relation between the spectrum of the specular diffuser and specular reflection.

where  $A$  is the complex amplitude,  $\mathbf{k}$  is the wave vector of magnitude  $|\mathbf{k}| = 2\pi/\lambda$ , and  $\mathbf{r} = (x \ y \ z)^t$  is the position vector. According to Eq. (7.10),  $g$  can therefore be interpreted as a superposition of plane waves of complex amplitude  $G(f_x, f_y)$  propagating with wave vector  $\mathbf{k}$  given by

$$\mathbf{k} = 2\pi \begin{pmatrix} f_x \\ f_y \\ \sqrt{\lambda^{-2} - f_x^2 - f_y^2} \end{pmatrix}. \quad (7.12)$$

In the diffuse-only version of the proposed method,  $\phi(x, y)$  is set to a uniform random value, and  $g$  has therefore a broadband spectrum, as shown in Figure 7.2a. This means that  $g$  is composed by an infinity of plane waves propagating in all the directions in space. As a consequence, each 3D scene point spreads light equally in all the directions, as shown in Figure 7.2b.

## 7.2.2 Creation of a new phase distribution

To render specular reflections, scene points should be initialized with a new phase distribution able to concentrate the energy of emitted light around a given direction, as shown in Figure 7.3b.

We consider the direction of emitted light given by angles  $\alpha$  and  $\beta$  around the vertical and horizontal axes, respectively. We call  $g'_{\alpha, \beta}$  the diffuser which scatters light around this direction. Therefore, its Fourier spectrum should be band-limited and shifted from the origin by  $(\Delta f_x, \Delta f_y)$ , as shown in Figure 7.3a. According to the grating equation [53], the shift in

frequencies is given by

$$\Delta f_x = \frac{\sin(\alpha)}{\lambda} \quad \text{and} \quad \Delta f_y = \frac{\sin(\beta)}{\lambda}. \quad (7.13)$$

We thus multiply  $G$  by a 2D Gaussian window to get a band-limited Fourier spectrum given by

$$G'_{\alpha,\beta}(f_x, f_y) = G(f_x, f_y) \exp \left[ -\frac{(f_x - \Delta f_x)^2 + (f_y - \Delta f_y)^2}{2\sigma^2} \right], \quad (7.14)$$

where  $\sigma$  is an adjustable parameter which controls the bandwidth of  $G'_{\alpha,\beta}$  and therefore the diffuseness of each scene point. This parameter depends on the wavelength of light.

Since  $|\mathcal{F}^{-1} \{G'_{\alpha,\beta}\}| \neq 1$ , we define the new phase distribution as

$$\phi'_{\alpha,\beta}(x, y) = \arg \{ \mathcal{F}^{-1} \{G'_{\alpha,\beta}\}(x, y) \}, \quad (7.15)$$

where  $\arg\{C\}$  is the phase of complex number  $C$ . The new diffuser is therefore given by

$$g'_{\alpha,\beta}(x, y) = \exp [j\phi'_{\alpha,\beta}(x, y)]. \quad (7.16)$$

Since a new diffuser has to be computed for every possible values of  $\alpha$  and  $\beta$ , this process is time-consuming. To reduce the computational load, we first compute  $g'_{0,0}$  using Eq. (7.16). Then, the other diffusers  $g'_{\alpha,\beta}$  are approximated by simply multiplying  $g'_{0,0}$  by a plane wave propagating in the direction given by angles  $\alpha$  and  $\beta$ , such that

$$g'_{\alpha,\beta}(x, y) = g'_{0,0}(x, y) \exp [j2\pi(x\Delta f_x + y\Delta f_y)]. \quad (7.17)$$

### 7.2.3 Overall algorithm

To generalize it to the rendering of specular reflections, the proposed method is modified as follows. First, 2D-plus-depth projections of the scene are taken from different perspective viewpoints, as described before. Then, the 3D scene geometry is reconstructed from MVD data as a layered point cloud. However, this time, the complex wave  $o_d$  emitted by each depth layer  $d$  is given by

$$o_d(x, y) = \sum_{\substack{i,j,u,v \\ D_{i,j}(u,v)=d}} \sqrt{I_{i,j}(u, v)} g'_{\theta_u, \theta_v}(x, y) \exp \left[ -\frac{(x - x_{i,u})^2 + (y - y_{j,v})^2}{2s^2} \right], \quad (7.18)$$

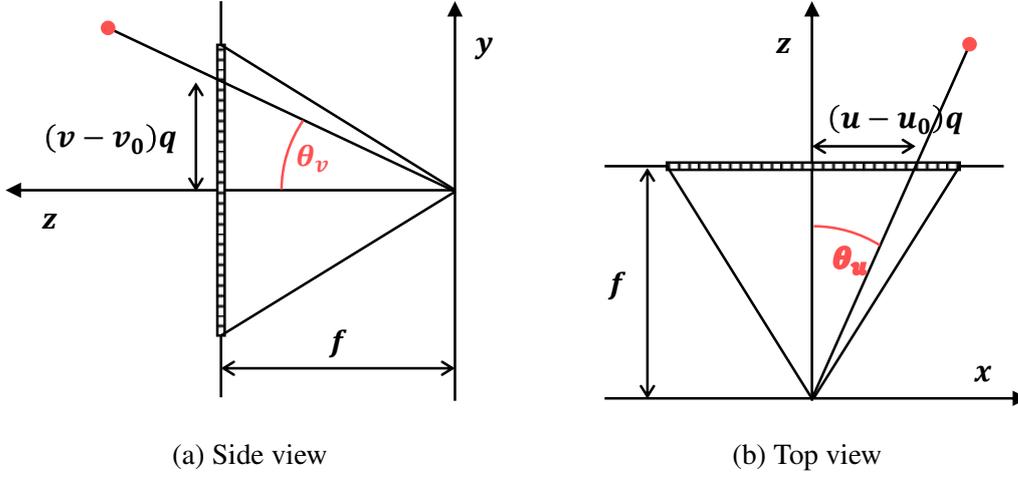


Fig. 7.4 Relation between the pixels coordinates of a point and the direction from which its intensity has been captured by the camera.

where

$$\theta_u = \arctan\left(\frac{(u - u_0)q}{f}\right) \quad \text{and} \quad \theta_v = \arctan\left(\frac{(v - v_0)q}{f}\right) \quad (7.19)$$

define the direction from which the 3D point intensity has been captured by the camera, as shown in Figure 7.4. Finally, light scattered by the scene is numerically propagated towards the hologram plane and encoded into the CGH, as described before.

## 7.3 Experimental results and discussion

The proposed method was implemented in C++/CUDA on a PC system employing an Intel Core i7-4930K CPU operating at 3.40 GHz, a main memory of 16 GB, three GPUs NVIDIA GeForce GTX 780Ti, and an operating system of Microsoft Windows 8.

### 7.3.1 Input 3D scenes

Table 7.1 shows the hologram synthesis parameters used for the experiments. We used Unity3D software to capture the MVD projections of a highly reflective car. In this software, we set up  $(9 \times 9)$  virtual cameras with resolution  $(1000 \times 1000)$ , focal length  $f = 2.0\text{mm}$  and pixel pitch  $q = 4.0\mu\text{m}$ . The distance between the cameras was set to  $d_x = d_y = 0.3\text{mm}$  along the horizontal and vertical axes and the scene was located between  $z_{\min} = 2.0\text{mm}$  and  $z_{\max} = 5.0\text{mm}$ .

For the 3D scene geometry reconstruction step, we set the standard deviation  $s$  of the 3D points' Gaussian distribution to be equal to  $2.5\mu\text{m}$ . The points' diffuseness parameter

Table 7.1 Hologram synthesis parameters used for the experiments.

MVD capture parameters		Hologram parameters	
Number of cameras	(9, 9)	Hologram resolution	(8192 × 8192)
Camera resolution	(1000, 1000)	Pixel pitch	0.36 $\mu$ m
Distance between cameras	(0.3mm, 0.3mm)	Red wavelength	640nm
Focal length	2.0mm	Green wavelength	532nm
Pixel pitch	4.0 $\mu$ m	Blue wavelength	473nm
Scene depth extent	(2.0mm, 5.0mm)		

$\sigma$  was set to 94, 224Hz, 113, 353Hz and 127, 492Hz for the Red, Green and Blue channels, respectively. Finally, the hologram is sampled on a regular 2D grid of resolution (8192×8192) with pixel pitch  $p = 0.36\mu\text{m}$ , and the wavelengths are set to 640nm, 532nm and 473nm for the Red, Green and Blue channels, respectively. In the following, we compare the proposed method presented in Section 7.2 with its diffuse-only version, presented in Section 7.1.

### 7.3.2 Analysis of the visual quality of the reconstructed images

Figure 7.5 shows the scene images numerically reconstructed from the CGHs patterns generated by the diffuse-only version of the method (first row) and by the proposed method (second row), captured from different viewpoints. For the sake of comparison, Computer Graphics images synthesized from the same viewpoints are shown on the third row. These experimental results show that both the proposed method and its diffuse-only version are able to produce correct motion parallax and occlusion effect, even if only a few MVD projections of the scene are used as input data.

To demonstrate the capability of reproducing specular reflections, we compare the numerically reconstructed scene images to their corresponding Computer Graphics images. As shown in the first row, the diffuse-only version of the method is unable to reproduce specular reflections. Indeed, the intensity of light reflected by the car hood remains the same in all the directions, whereas it should vary according to the viewpoint. On the other hand, specular reflections are reproduced without any visible artifact using the proposed method, as shown in the second row. Finally, Figure 7.6 shows the scene images numerically reconstructed from the CGH generated by the proposed method, focused on different depths. As shown in this figure, when focusing on the headlights, the rear of the car is blurred, and vice versa. These results show that the proposed method produces an accurate accommodation cue with continuous depth change.

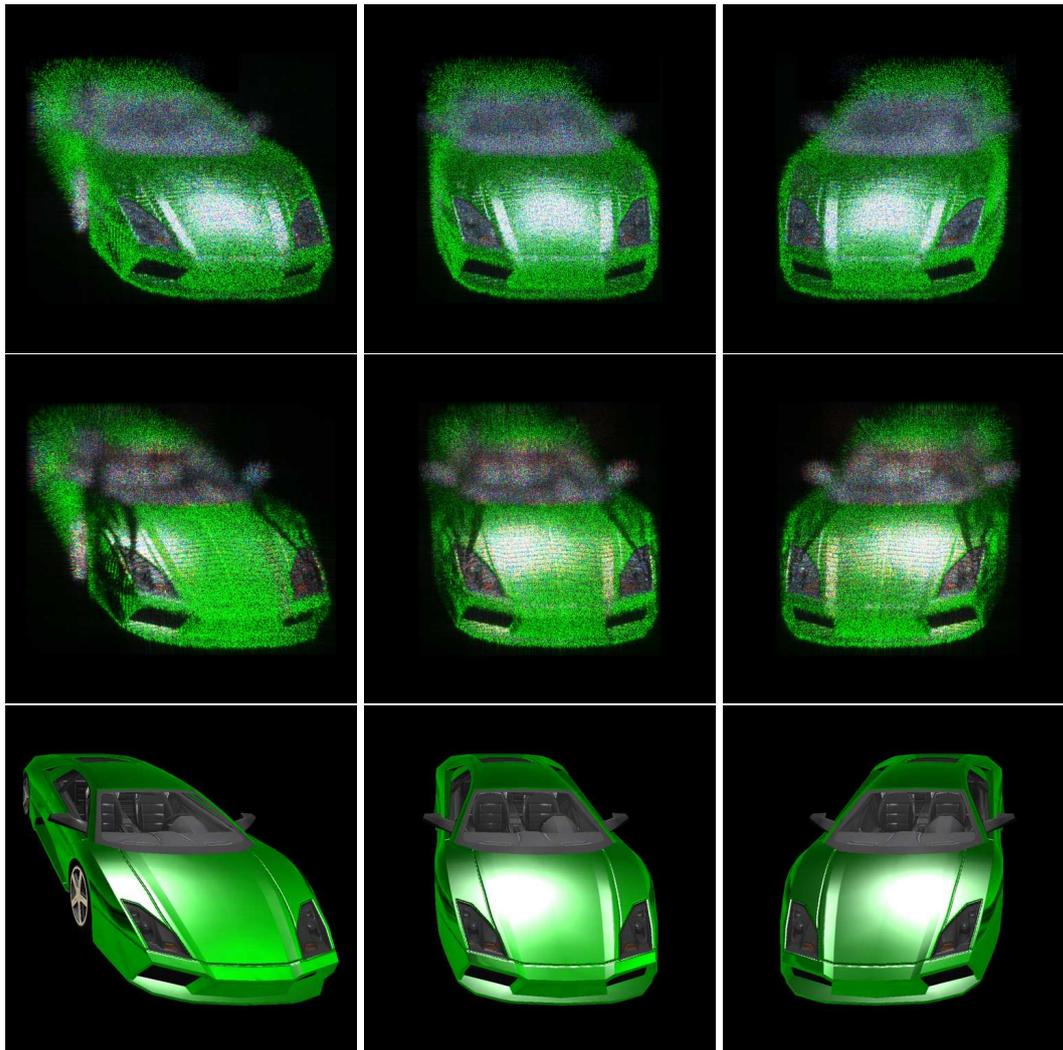


Fig. 7.5 Scene images numerically reconstructed from the CGHs generated by the diffuse-only version of the method (first row) and by the proposed method (second row), captured from different viewpoints. For the sake of comparison, Computer Graphics images synthesized from the same viewpoints are shown on the third row.

### 7.3.3 Analysis of the hologram calculation time

To evaluate the performance of the proposed method, we generated six holograms of the reflective car scene with different numbers of MVD projections. Figure 7.7 shows the CGH calculation time depending on the number of MVD projections using the proposed method and its diffuse-only version.

The third step does not depend on the MVD projections and is exactly the same for the two methods. As a consequence, its calculation time remains constant for both of them. On the contrary, since the number of reconstructed scene points depends on the number of MVD

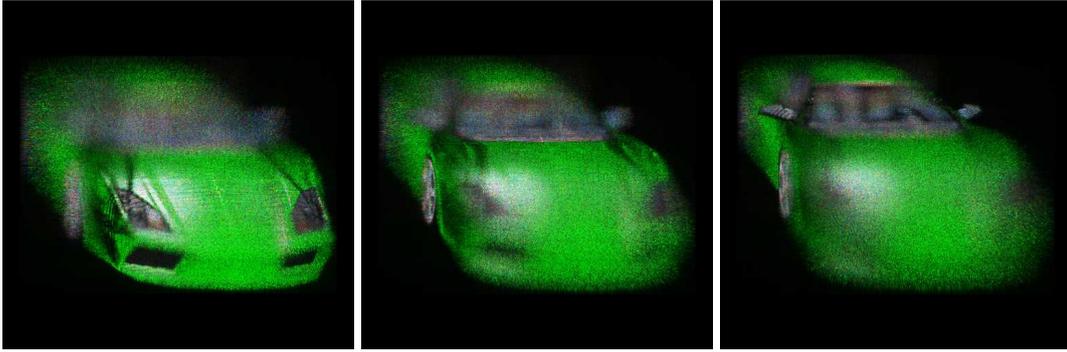


Fig. 7.6 Scene images numerically reconstructed from the CGH generated by the proposed method, focused on different depths.

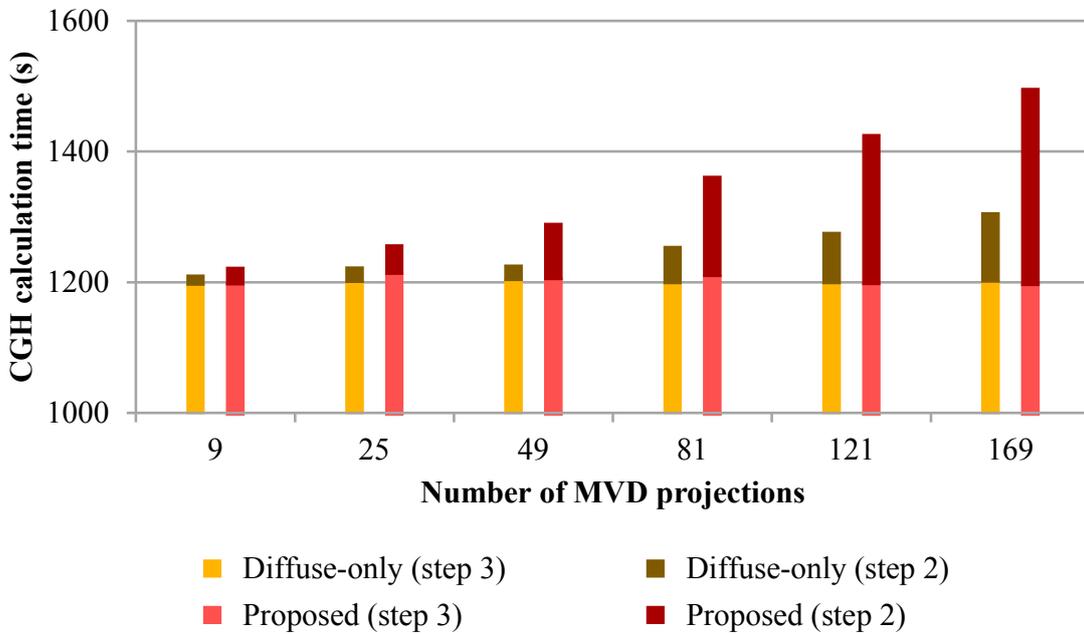


Fig. 7.7 CGH calculation time depending on the number of MVD projections.

projections, the calculation time of the second step increases linearly with it. Moreover, it is always more important for the proposed method than for its diffuse-only version. This is due to the fact that the computation of  $g'_{\alpha,\beta}$  is more time-expensive than the computation of  $g$ . However, in all cases, most of the CGH calculation time is spent during the third step. As a consequence, the CGH calculation time of the proposed method has been increased only by 0.99%, 5.22%, 8.57% and 14.58% when the number of MVD projections is equal to 9, 49, 81 and 169, respectively, compared to the diffuse-only version.

## 7.4 Conclusion

In this chapter, we developed our last contribution, dedicated to the rendering of specular reflections when computing a hologram from MVD data. The proposed method consists of three steps. First, 2D-plus-depth projections of the scene are taken from different perspective viewpoints. Then, the 3D scene geometry is reconstructed from the MVD data as a layered point-cloud. To take into account specular reflections, each scene point is considered to emit light differently in all the directions. Finally, light scattered by the scene is numerically propagated towards the hologram plane and encoded into the CGH.

Experimental results reveal that the proposed method is able to produce motion parallax with occlusion effect and accurate accommodation cue even if only a few MVD projections of the scene are used as input data. Furthermore, specular reflections are accurately reproduced with a small computational overhead.

## **Part V**

# **Conclusion**



# Chapter 8

## Conclusions and future work

Thanks to its attractive visualization properties, CGH may have application in the field of videoconferencing or telepresence systems in the future. However, due to the huge amount of data to process, it is still very challenging for state-of-the-art methods to produce holograms of realistic 3D scenes with strong depth illusion in real-time.

The objective of this thesis was thus twofold: (1) investigate algorithms to reduce the hologram calculation time and (2) design new methods to enhance the realism and visual quality of displayed 3D scenes. The final goal was to propose a method able to produce holograms of realistic scenes which provide all the human depth cues with reduced computational complexity. Section 8.1 summarizes the main contributions of this thesis, and Section 8.2 addresses future work and potential improvements. Finally, a general discussion on CGH open issues and prospects is given in Section 8.3.

### 8.1 Achieved work

The contributions of this thesis are organized into two parts. The first part of this work aims at reducing the hologram calculation time to get closer to real-time computation. To this end, we designed and proposed two novel algorithms.

First, we designed an efficient hybrid method for the calculation of colorful holograms from synthetic 3D models or 2D-plus-depth data by combining two approaches which complement one another: the point-source and wave-field approaches. Contrary to state-of-the-art methods, which aimed at reducing the computational complexity of these approaches independently, our method uses them together and therefore takes advantages from both of them. In this method, the 3D scene is sliced into several depth layers parallel to the hologram plane. Then, depending on the number of scene points they contain, the complex wave scattered by each layer is computed using either a wave-field or a point-source approach.

Finally, the complex waves scattered by all the depth layers are summed up to obtain the CGH. Experimental results reveal that using this approach, the hologram calculation time can be reduced by more than 65% compared to the conventional point-source and wave-field methods. We presented this work at the IEEE International Conference on Image Processing (ICIP) in 2015 [46].

Secondly, we further accelerated this hybrid method by removing temporal redundancies between consecutive frames of a 2D-plus-depth video. For each video frame, the algorithm detects changes in the scene and updates the hologram of only affected scene points. Since only small regions of the hologram are updated at each video frame, this method allows the computational burden to be dramatically reduced, enabling the computation of colorful video holograms at 60 frames per second. We presented this work at the SPIE Optics + Photonics, Applications of Digital Image Processing conference in 2015 [48].

The second part of this work aims at improving the realism and visual quality of displayed scenes. To this end, we proposed two novel algorithms.

First, we improved the hybrid method to take into account occlusions between objects in the scene during the calculation of holograms from synthetic 3D models. To this end, we designed an efficient algorithm for light shielding between scene points and light waves, allowing the occlusion of wave-field layers by point-source layers and vice-versa. Experimental results revealed that occlusions are accurately reproduced using this method without significantly increasing the hologram calculation time of the original hybrid method. This work was protected by a granted patent [49]. The outlines of the method were presented at the Pacific Conference on Computer Graphics and Applications (Pacific Graphics) in 2015 [47], and its detailed description has been published in The Optical Society's Applied Optics journal [51].

Finally, we proposed a hologram computation method from MVD data with rendering of specular reflections. In this method, the 3D scene geometry is first reconstructed from the MVD data as a layered point-cloud, enabling the use of only a few perspective projections of the scene. Furthermore, to take into account specular reflections, each scene point is considered to emit light differently in all the directions. Finally, light scattered by the scene is numerically propagated towards the hologram plane to get the CGH. Experimental results showed that this method is able to provide all the human depth cues and accurate shading of the scene with reduced computational complexity. We presented this work at the IEEE International Workshop on Hot Topics in 3D (Hot3D) in 2016 [50].

To summarize, during this thesis we first investigated two algorithms to reduce the hologram calculation time. The first one is based on an efficient hybrid point-source/wave-field approach, while the second uses temporal redundancy removal to further accelerate

the calculation of video holograms. These methods enabled us to compute colorful video holograms at interactive rates. In the second part of this thesis, we improved these algorithms to enhance the realism and visual quality of displayed scenes without significantly increasing the hologram calculation time. We thus proposed two novel methods to take into account occlusions between objects in the scene and specular reflections efficiently. Thanks to these contributions, we were able to produce holograms of realistic scenes which provide all the human depth cues with reduced computational complexity. Furthermore, this work led to several publications and one patent, which are listed in Appendix A.

## 8.2 Future work

As stated before, the contributions of this thesis enable us to produce holograms of realistic scenes which provide all the human depth cues with reduced computational complexity. However, our work has some limitations which should be addressed in future work.

First of all, the proposed hybrid method and its extensions are limited to layered models, which is not the most realistic 3D scene representation. Indeed, even if the number of depth layers is set such that the separation between two consecutive layers remains invisible, this approximation of the 3D scene geometry can lead to leakage light passing through the gaps between two cross-section masks. To overcome this limitation, this approach should be generalized to polygon meshes. In this case, the polygons composing the scene could be classified into two categories: the point-source polygons, whose complex wave is computed using the point-source approach, and the wave-field polygons, whose complex wave is computed using propagation between tilted planes. This classification could be performed depending on the size and orientation of each polygon. Moreover, the light shielding technique developed in Chapter 6 should be generalized to polygon meshes.

Secondly, the proposed hybrid method with temporal redundancy removal was only tested with fully synthetic scenes. In this case, depth maps are perfect in terms of accuracy and alignment with intensity data. Moreover, the depth information is temporally coherent and stable, which enables efficient temporal redundancy removal from one frame to another. However, when capturing a natural scene with a 2D-plus-depth camera, depth information will inevitably be inaccurate. As a consequence, the sensitivity and performance of the proposed method to errors in depth data when using real cameras should be studied. Moreover, other data compression schemes should be evaluated to further reduce the CGH calculation time. For example, object-based and block-based motion compensation schemes could be investigated, and their computational complexities analyzed and compared to the proposed method.

Thirdly, further investigations are required to improve the proposed hologram synthesis method from MVD data. In particular, the influence of the 3D points' Gaussian distribution and diffuseness parameter on 3D scene realism and visual quality should be carefully evaluated. Some formulas should be derived to automatically determine the values of these parameters depending on the characteristics and arrangement of the 2D camera array. Finally, this method should also be evaluated using real-existing scenes captured under natural light.

### 8.3 Open issues and prospects

Thanks to its remarkable 3D visualization properties, Holography has gained a significant interest by the research community. In this context, the Joint Photographic Experts Group (JPEG) Committee has launched a new activity, JPEG PLENO, to standardize the representation and exchange of new imaging formats, including holographic data [1]. Furthermore, an open access database of reference computer-generated holograms has been set-up to enable researchers to evaluate and compare different codecs [12]. However, in addition to the computational issues we tackle in this thesis, some technical challenges remain to be investigated.

First, in order to compute holograms of real existing scenes, it is necessary to use a 3D capture set-up. However, current 3D capture systems still suffer from technical challenges which prevent them from being widely used in CGH applications. Indeed, these systems are either too bulky, too expensive, or not accurate enough for immersive 3D visualization. For instance, depth cameras are cheap and portable, but their accuracy in 3D measurement is relatively low, which may result in unwanted artifacts during the visualization. On the other hand, setting up a camera array enables the acquisition of accurate data, but these systems are generally expensive and not easily transportable. Finally, while being a very promising technology, the capture angle of consumer plenoptic cameras is not wide enough to be used in full parallax systems.

Another challenge is to display large full-color CGHs with wide viewing angle on SLMs. Indeed, due to their limited resolution and to their pixel pitches, current SLMs are not able to display CGHs larger than a few square centimeters and with viewing zone angle wider than a few degrees. As a consequence, to perform subjective quality assessment of holograms, other plotting techniques have to be investigated [4]. Nevertheless, with the growing interest in Virtual Reality (VR) and Augmented Reality (AR), several prototypes of Holographic Head-Mounted Displays (HMD) have recently been designed [121, 212]. In this case, the difficulty due to the limited size of current SLMs is alleviated by placing them close to

the eyes of the viewer. Moreover, since motion parallax is provided according to the head movements of the viewer, the viewing zone angle of current SLMs is no longer an issue.

Finally, the lack of objective quality assessments measures suitable to holographic data is still an open issue. State-of-the-art papers often use conventional measures designed for 2D images, such as PSNR or Structural Similarity Index (SSIM) and evaluate either the quality of numerically reconstructed images or of the hologram itself [4]. However, since holographic data have very different signal properties compared to 2D images and videos, other quality assessment measures more appropriate for holographic data should be investigated.



## **Part VI**

### **References and Appendix**



# References

- [1] JPEG - JPEG PLENO Abstract and Executive Summary.
- [2] D. Abookasis and J. Rosen. Computer-generated holograms of three-dimensional objects synthesized from their multiple angular viewpoints. *Journal of the Optical Society of America A*, 20(8):1537–1545, Aug. 2003.
- [3] D. Abookasis and J. Rosen. Three types of computer-generated hologram synthesized from multiple angular viewpoints of a three-dimensional scene. *Applied Optics*, 45(25):6533–6538, Sept. 2006.
- [4] A. Ahar, D. Blinder, T. Bruylants, C. Schretter, A. Munteanu, and P. Schelkens. Subjective quality assessment of numerically reconstructed compressed holograms. In *Applications of Digital Image Processing XXXVIII*, volume Proc. SPIE 9599, pages 95990K–95990K–15, Sept. 2015.
- [5] L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson. Computer generated holography using parallel commodity graphics hardware. *Optics express*, 14(17):7636–7641, Aug. 2006.
- [6] L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson. Computer generated holograms from three dimensional meshes using an analytic light transport model. *Applied Optics*, 47(10):1567–1574, Apr. 2008.
- [7] V. Arrizón. Complex modulation with a twisted-nematic liquid-crystal spatial light modulator: double-pixel approach. *Optics Letters*, 28(15):1359–1361, Aug. 2003.
- [8] V. Arrizón, G. Méndez, and D. Sánchez-de La-Llave. Accurate encoding of arbitrary complex fields with amplitude-only liquid crystal spatial light modulators. *Optics Express*, 13(20):7913–7927, Oct. 2005.
- [9] J. Barabas, S. Jolly, D. E. Smalley, and J. Bove. Diffraction specific coherent panoramagrams of real scenes. In *Practical Holography XXV: Materials and Applications*, volume Proc. SPIE 7957, pages 795702–795702–7, Feb. 2011.
- [10] M. Bayraktar and M. Özcan. Method to calculate the far field of three-dimensional objects for computer-generated holography. *Applied Optics*, 49(24):4647–4654, Aug. 2010.
- [11] P. Birch, R. Young, C. Chatwin, M. Farsari, D. Budgett, and J. Richardson. Fully complex optical modulation with an analogue ferroelectric liquid crystal spatial light modulator. *Optics Communications*, 175(4-6):347–352, Mar. 2000.

- [12] D. Blinder, A. Ahar, A. Symeonidou, Y. Xing, T. Bruylants, C. Schreites, B. Pesquet-Popescu, F. Dufaux, A. Munteanu, and P. Schelkens. Open access database for experimental validations of holographic compression engines. In *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 1–6, May 2015.
- [13] D. Blinder, T. Bruylants, H. Ottevaere, A. Munteanu, and P. Schelkens. JPEG 2000-based compression of fringe patterns for digital holographic microscopy. *Optical Engineering*, 53(12):123102–123102, Dec. 2014.
- [14] J. F. Blinn. Models of Light Reflection for Computer Synthesized Pictures. In *4th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '77*, pages 192–198, 1977.
- [15] B. R. Brown and A. W. Lohmann. Complex Spatial Filtering with Binary Masks. *Applied Optics*, 5(6):967–969, June 1966.
- [16] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, Feb. 2006.
- [17] M. P. Chang and O. K. Ersoy. Iterative interlacing error diffusion for synthesis of computer-generated holograms. *Applied Optics*, 32(17):3122–3129, June 1993.
- [18] N. Chen, J.-H. Park, and N. Kim. Parameter analysis of integral Fourier hologram and its resolution enhancement. *Optics Express*, 18(3):2152–2167, Feb. 2010.
- [19] N. Chen, J. Yeom, J.-H. Jung, J.-H. Park, and B. Lee. Resolution comparison between integral-imaging-based hologram synthesis methods using rectangular and hexagonal lens arrays. *Optics Express*, 19(27):26917–26927, Dec. 2011.
- [20] R. H.-Y. Chen and T. D. Wilkinson. Computer generated hologram from point cloud using graphics processor. *Applied Optics*, 48(36):6841–6850, Dec. 2009.
- [21] R. H.-Y. Chen and T. D. Wilkinson. Computer generated hologram with geometric occlusion using GPU-accelerated depth buffer rasterization for three-dimensional display. *Applied optics*, 48(21):4246–4255, July 2009.
- [22] B. B. Chhetri, S. Yang, and T. Shimomura. Stochastic approach in the efficient design of the direct-binary-search algorithm for hologram synthesis. *Applied Optics*, 39(32):5956–5964, Nov. 2000.
- [23] J. Cho, H. Kim, J. Yeom, G. Li, and B. Lee. Rendering of reflective surfaces in polygon based computer generated holograms. In *Imaging and Applied Optics 2015*, OSA Technical Digest, page JT5A.21, June 2015.
- [24] S. Cho, B.-K. Ju, N.-Y. Kim, and M.-C. Park. One-eighth look-up table method for effectively generating computer-generated hologram patterns. *Optical Engineering*, 53(5):054108–054108, May 2014.
- [25] R. L. Cook and K. E. Torrance. A Reflectance Model for Computer Graphics. *ACM Trans. Graph.*, 1(1):7–24, Jan. 1982.

- [26] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.
- [27] J. Cutting and P. Vishton. Perceiving layout and knowing distances: the integration, relative potency and contextual use of different information about depth. In W. Epstein and S. Rogers, editors, *Handbook of perception and Cognition.*, volume 5: Perception of Space and Motion, pages 69–117. 1995.
- [28] E. Darakis and T. J. Naughton. Compression of digital hologram sequences using MPEG-4. In *Holography: Advances and Modern Trends*, volume Proc. SPIE 7358, pages 735811–735811–8, May 2009.
- [29] E. Darakis and J. J. Soraghan. Use of Fresnelets for Phase-Shifting Digital Hologram Compression. *IEEE Transactions on Image Processing*, 15(12):3804–3811, Dec. 2006.
- [30] M. Deering. High Resolution Virtual Reality. In *19th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '92, pages 195–202, 1992.
- [31] N. Delen and B. Hooker. Free-space beam propagation between arbitrarily oriented planes based on full diffraction theory: a fast Fourier transform approach. *Journal of the Optical Society of America A*, 15(4):857–867, Apr. 1998.
- [32] N. A. Dodgson. Analysis of the viewing zone of multiview autostereoscopic displays. In *Stereoscopic Displays and Virtual Reality Systems IX*, volume Proc. SPIE 4660, pages 254–265, May 2002.
- [33] X.-B. Dong, H.-M. Choi, M.-W. Kwon, S.-C. Kim, and E.-S. Kim. Fast generation of video hologram patterns by use of motion vectors of three-dimensional objects. In *Practical Holography XXVII: Materials and Applications*, volume Proc. SPIE 8644, pages 864418–864418–8, Mar. 2013.
- [34] X.-B. Dong, S.-C. Kim, and E.-S. Kim. MPEG-based novel look-up table for rapid generation of video holograms of fast-moving three-dimensional objects. *Optics Express*, 22(7):8047, Apr. 2014.
- [35] X.-B. Dong, S.-C. Kim, and E.-S. Kim. Three-directional motion compensation-based novel-look-up-table for video hologram generation of three-dimensional objects freely maneuvering in space. *Optics Express*, 22(14):16925, July 2014.
- [36] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, Apr. 2006.
- [37] F. Dufaux, Y. Xing, B. Pesquet-Popescu, and P. Schelkens. Compression of digital holographic data: an overview. In *Applications of Digital Image Processing XXXVIII*, volume Proc. SPIE 9599, pages 95990I–95990I–11, Sept. 2015.
- [38] O. K. Ersoy, J. Y. Zhuang, and J. Brede. Iterative interlacing approach for synthesis of computer-generated holograms. *Applied Optics*, 31(32):6894–6901, Nov. 1992.
- [39] J. R. Fienup. Iterative Method Applied To Image Reconstruction And To Computer-Generated Holograms. *Optical Engineering*, 19(3):193297–193297–, June 1980.

- [40] C. Frère and D. Leseberg. Large objects reconstructed from computer-generated holograms. *Applied Optics*, 28(12):2422–2425, June 1989.
- [41] T. Fujii and H. Yoshikawa. Improvement of Hidden-Surface Removal for Computer-Generated Holograms from CG. In *Adaptive Optics: Analysis and Methods/Computational Optical Sensing and Imaging/Information Photonics/Signal Recovery and Synthesis Topical Meetings on CD-ROM*, OSA Technical Digest, page DWB3, June 2007.
- [42] D. Gabor. A new microscopic principle. *Nature*, 161(4098):777, 1948.
- [43] S. Ganci. Fourier diffraction through a tilted slit. *European Journal of Physics*, 2(3):158, 1981.
- [44] C. Gao, J. Liu, X. Li, G. Xue, J. Jia, and Y. Wang. Accurate compressed look up table method for CGH in 3d holographic display. *Optics Express*, 23(26):33194, Dec. 2015.
- [45] R. Gerchberg and O. Saxton. A practical algorithm for the determination of the phase from image and diffraction plane pictures. *Optik*, 35:237–246, Nov. 1971.
- [46] A. Gilles, P. Gioia, R. Cozot, and L. Morin. Complex modulation computer-generated hologram by a fast hybrid point-source/wave-field approach. In *IEEE International Conference on Image Processing (ICIP) 2015*, pages 4962–4966, Sept. 2015.
- [47] A. Gilles, P. Gioia, R. Cozot, and L. Morin. Complex modulation computer-generated hologram with occlusion effect by a fast hybrid point-source/wave-field approach. In *Pacific Conference on Computer Graphics and Applications (Pacific Graphics) 2015*, Oct. 2015.
- [48] A. Gilles, P. Gioia, R. Cozot, and L. Morin. Fast generation of complex modulation video holograms using temporal redundancy compression and hybrid point-source/wave-field approaches. In *Applications of Digital Image Processing XXXVIII*, volume Proc. SPIE 9599, pages 95990J–95990J–14, Sept. 2015.
- [49] A. Gilles, P. Gioia, R. Cozot, and L. Morin. Procédé de génération numérique d’un hologramme, dispositif et programme d’ordinateur associés, May 2015.
- [50] A. Gilles, P. Gioia, R. Cozot, and L. Morin. Computer-generated hologram from Multiview-plus-Depth data considering specular reflections. In *IEEE International Workshop on Hot Topics in 3D (Hot3D)*, July 2016.
- [51] A. Gilles, P. Gioia, R. Cozot, and L. Morin. Hybrid approach for fast occlusion processing in computer-generated hologram calculation. *Applied Optics*, 55(20):5459–5470, July 2016.
- [52] E. B. Goldstein. *Sensation and Perception*. Cengage Learning, Feb. 2009.
- [53] J. W. Goodman. *Introduction to Fourier Optics*. Roberts and Company Publishers, Englewood, Colo, 3rd edition, 2005.
- [54] J. Hahn, H. Kim, K. Choi, and B. Lee. Real-time digital holographic beam-shaping system with a genetic feedback tuning loop. *Applied optics*, 45(5):915–924, Feb. 2006.

- [55] P. M. Hirsch, J. A. Jordan, and B. J. Lesem. Method of making an object independent diffuser, 1971.
- [56] D. M. Hoffman, A. R. Girshick, K. Akeley, and M. S. Banks. Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision*, 8(3):33, Mar. 2008.
- [57] K. Hosoyachi, K. Yamaguchi, T. Ichikawa, and Y. Sakamoto. Precalculation method using spherical basic object light for computer-generated hologram. *Applied Optics*, 52(1):A33–A44, Jan. 2013.
- [58] M.-L. Hsieh, M.-L. Chen, and C.-J. Cheng. Improvement of the complex modulated characteristic of cascaded liquid crystal spatial light modulators by using a novel amplitude compensated technique. *Optical Engineering*, 46(7):070501–070501–3, July 2007.
- [59] Y. Ichihashi, H. Nakayama, T. Ito, N. Masuda, T. Shimobaba, A. Shiraki, and T. Sugie. HORN-6 special-purpose clustered computing system for electroholography. *Optics Express*, 17(16):13895–13903, Aug. 2009.
- [60] T. Ichikawa, Y. Sakamoto, A. Subagyo, and K. Sueoka. Calculation method of reflectance distributions for computer-generated holograms using the finite-difference time-domain method. *Applied Optics*, 50(34):H211–H219, Dec. 2011.
- [61] T. Ichikawa, Y. Sakamoto, A. Subagyo, and K. Sueoka. A method of calculating reflectance distributions for CGH with FDTD using the structure of actual surfaces. In *Practical Holography XXV: Materials and Applications*, volume Proc. SPIE 7957, pages 795707–795707–6, Feb. 2011.
- [62] T. Ichikawa, T. Yoneyama, and Y. Sakamoto. CGH calculation with the ray tracing method for the Fourier transform optical system. *Optics Express*, 21(26):32019–32031, Dec. 2013.
- [63] D. Im, J. Cho, J. Hahn, B. Lee, and H. Kim. Accelerated synthesis algorithm of polygon computer-generated holograms. *Optics Express*, 23(3):2863–2871, Feb. 2015.
- [64] T. Ito, N. Masuda, K. Yoshimura, A. Shiraki, T. Shimobaba, and T. Sugie. Special-purpose computer HORN-5 for a real-time electroholography. *Optics Express*, 13(6):1923–1932, Mar. 2005.
- [65] B. K. Jennison, D. W. Sweeney, and J. P. Allebach. Efficient design of direct-binary-search computer-generated holograms. *JOSA A*, 8(4):652–660, Apr. 1991.
- [66] K.-M. Jeong, H.-S. Kim, S.-I. Hong, S.-K. Lee, N.-Y. Jo, Y.-S. Kim, H.-G. Lim, and J.-H. Park. Acceleration of integral imaging based incoherent Fourier hologram capture using graphic processing unit. *Optics Express*, 20(21):23735–23743, Oct. 2012.
- [67] J. Jia, J. Liu, G. Jin, and Y. Wang. Fast and effective occlusion culling for 3d holographic displays by inverse orthographic projection with low angular sampling. *Applied Optics*, 53(27):6287–6293, Sept. 2014.

- [68] J. Jia, Y. Wang, J. Liu, X. Li, Y. Pan, Z. Sun, B. Zhang, Q. Zhao, and W. Jiang. Reducing the memory usage for effective computer-generated hologram calculation using compressed look-up table in full-color holographic display. *Applied Optics*, 52(7):1404–1412, Mar. 2013.
- [69] E. G. Johnson and M. A. G. Abushagur. Microgenetic-algorithm optimization methods applied to dielectric gratings. *JOSA A*, 12(5):1152–1160, May 1995.
- [70] J. L. Juárez-Pérez, A. Olivares-Pérez, and L. R. Berriel-Valdos. Nonredundant calculations for creating digital Fresnel holograms. *Applied Optics*, 36(29):7437–7443, Oct. 1997.
- [71] T. Kakue, N. Masuda, Y. Endo, R. Hirayama, N. Okada, T. Shimobaba, and T. Ito. Special-purpose computer for real-time reconstruction of holographic motion picture. In *2013 International Conference on Optical Instruments and Technology: Optical Systems and Modern Optoelectronic Instruments*, volume Proc. SPIE 9042, pages 90420B–90420B–7, Dec. 2013.
- [72] H. Kang, T. Fujii, T. Yamaguchi, and H. Yoshikawa. Compensated phase-added stereogram for real-time holographic display. *Optical Engineering*, 46(9):095802–095802–11, Sept. 2007.
- [73] H. Kang, E. Stoykova, and H. Yoshikawa. Fast phase-added stereogram algorithm for generation of photorealistic 3d content. *Applied Optics*, 55(3):A135, Jan. 2016.
- [74] H. Kang, T. Yamaguchi, and H. Yoshikawa. Accurate phase-added stereogram to improve the coherent stereogram. *Applied Optics*, 47(19):D44–D54, July 2008.
- [75] H. Kang, F. Yaras, L. Onural, and H. Yoshikawa. Real-Time Fringe Pattern Generation with High Quality. In *Advances in Imaging*, OSA Technical Digest, page DTuB7, Apr. 2009.
- [76] B. Katz, N. T. Shaked, and J. Rosen. Synthesizing computer generated holograms with reduced number of perspective projections. *Optics Express*, 15(20):13250–13255, Oct. 2007.
- [77] H. Kim, J. Hahn, and B. Lee. Mathematical modeling of triangle-mesh-modeled three-dimensional surface objects for digital holography. *Applied Optics*, 47(19):D117–D127, July 2008.
- [78] M.-S. Kim, G. Baasantseren, N. Kim, and J.-H. Park. Hologram Generation of 3d Objects Using Multiple Orthographic View Images. *Journal of the Optical Society of Korea*, 12(4):269–274, Dec. 2008.
- [79] S.-C. Kim, W.-Y. Choe, and E.-S. Kim. Accelerated computation of hologram patterns by use of interline redundancy of 3-D object images. *Optical Engineering*, 50(9):091305–091305–10, Aug. 2011.
- [80] S.-C. Kim, X.-B. Dong, M.-W. Kwon, and E.-S. Kim. Fast generation of video holograms of three-dimensional moving objects using a motion compensation-based novel look-up table. *Optics Express*, 21(9):11568–11584, May 2013.

- [81] S.-C. Kim and E.-S. Kim. Effective generation of digital holograms of three-dimensional objects using a novel look-up table method. *Applied Optics*, 47(19):D55–D62, July 2008.
- [82] S.-C. Kim and E.-S. Kim. Fast computation of hologram patterns of a 3d object using run-length encoding and novel look-up table methods. *Applied Optics*, 48(6):1030–1041, Feb. 2009.
- [83] S.-C. Kim, J.-M. Kim, and E.-S. Kim. Effective memory reduction of the novel look-up table with one-dimensional sub-principle fringe patterns in computer-generated holograms. *Optics Express*, 20(11):12021–12034, May 2012.
- [84] S.-C. Kim, J.-W. Moon, D.-H. Lee, K.-C. Son, and E.-S. Kim. Holographic full-color 3d display system using color-LCoS spatial light modulator. In *Practical Holography XIX: Materials and Applications*, volume Proc. SPIE 5742, pages 223–233, May 2005.
- [85] S.-C. Kim, J.-H. Yoon, and E.-S. Kim. Fast generation of three-dimensional video holograms by combined use of data compression and lookup table techniques. *Applied Optics*, 47(32):5986–5995, Nov. 2008.
- [86] S.-G. Kim, B. Lee, and E.-S. Kim. Removal of bias and the conjugate image in incoherent on-axis triangular holography and real-time reconstruction of the complex hologram. *Applied Optics*, 36(20):4784–4791, July 1997.
- [87] Y. Kim, E. Stoykova, H. Kang, S. Hong, J. Park, J. Park, and J. Hong. Seamless full color holographic printing method based on spatial partitioning of SLM. *Optics Express*, 23(1):172–182, Jan. 2015.
- [88] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May 1983.
- [89] J. S. Kollin, S. A. Benton, and M. L. Jepsen. Real-Time Display Of 3-D Computed Holograms By Scanning The Image Of An Acousto-Optic Modulator. In *Holographic Optics II: Principles and Applications*, volume Proc. SPIE 1136, pages 178–185, Oct. 1989.
- [90] A. Kondoh and K. Matsushima. Hidden Surface Removal in Full-parallax CGHs by Silhouette Approximation. *Syst. Comput. Japan*, 38(6):53–61, June 2007.
- [91] T. Kurihara and Y. Takaki. Shading of holographic reconstructed image by two-dimensional amplitude modulation of zone plates. In *Practical Holography XXVI: Materials and Applications*, volume Proc. SPIE 8281, pages 82810H–82810H–6, Feb. 2012.
- [92] K. Kushimoto and Y. Sakamoto. Computer-Generated Hologram Calculated from Multi-View Images of Real Existing Objects. In *Adaptive Optics: Analysis and Methods/Computational Optical Sensing and Imaging/Information Photonics/Signal Recovery and Synthesis Topical Meetings on CD-ROM*, OSA Technical Digest, page DWB6, June 2007.

- [93] M.-W. Kwon, S.-C. Kim, and E.-S. Kim. Graphics processing unit-based implementation of a one-dimensional novel-look-up-table for real-time computation of Fresnel hologram patterns of three-dimensional objects. *Optical Engineering*, 53(3):035103–035103, Mar. 2014.
- [94] M.-W. Kwon, S.-C. Kim, S.-E. Yoon, Y.-S. Ho, and E.-S. Kim. Object tracking mask-based NLUT on GPUs for real-time generation of holographic videos of three-dimensional scenes. *Optics Express*, 23(3):2101, Feb. 2015.
- [95] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, Feb. 1994.
- [96] W. Lee, D. Im, J. Paek, J. Hahn, and H. Kim. Semi-analytic texturing algorithm for polygon computer-generated holograms. *Optics Express*, 22(25):31180–31191, Dec. 2014.
- [97] E. N. Leith and J. Upatnieks. Reconstructed Wavefronts and Communication Theory. *Journal of the Optical Society of America*, 52(10):1123–1128, Oct. 1962.
- [98] E. N. Leith and J. Upatnieks. Wavefront Reconstruction with Diffused Illumination and Three-Dimensional Objects. *Journal of the Optical Society of America*, 54(11):1295–1301, Nov. 1964.
- [99] D. Leseberg and C. Frère. Computer-generated holograms of 3-D objects composed of tilted planar segments. *Applied Optics*, 27(14):3020–3024, July 1988.
- [100] Y. Li, D. Abookasis, and J. Rosen. Computer-Generated Holograms of Three-Dimensional Realistic Objects Recorded Without Wave Interference. *Applied Optics*, 40(17):2864–2870, June 2001.
- [101] G. Lippmann. Épreuves réversibles donnant la sensation du relief. *J. Phys. Theor. Appl.*, 7(1):821–825, 1908.
- [102] J.-P. Liu, W.-Y. Hsieh, T.-C. Poon, and P. Tsang. Complex Fresnel hologram display using a single SLM. *Applied Optics*, 50(34):H128, Dec. 2011.
- [103] Y. Liu, J. Dong, Y. Pu, B. Chen, H. He, Z. Deng, and H. Wang. A fast analytical algorithm for generating CGH of 3d scene. In *Practical Holography XXIV: Materials and Applications*, volume Proc. SPIE 7619, pages 76190N–76190N–8, Feb. 2010.
- [104] Y. Liu, J. Dong, Y. Pu, B. Chen, H. He, and H. Wang. High-speed full analytical holographic computations for true-life scenes. *Optics Express*, 18(4):3345–3351, Feb. 2010.
- [105] A. W. Lohmann. Three-dimensional properties of wave-fields. *Optik*, 51:105–107, 1978.
- [106] M. Lucente. *Diffraction-specific fringe computation for electro-holography*. PhD thesis, Massachusetts Institute of Technology, 1994.

- [107] M. E. Lucente. Interactive computation of holograms using a look-up table. *Journal of Electronic Imaging*, 2(1):28–34, Jan. 1993.
- [108] Z. M. A. Lum, X. Liang, Y. Pan, R. Zheng, and X. Xu. Increasing pixel count of holograms for three-dimensional holographic display by optical scan-tiling. *Optical Engineering*, 52(1):015802–015802, Jan. 2013.
- [109] K. Maeno, N. Fukaya, O. Nishikawa, K. Sato, and T. Honda. Electro-holographic display using 15mega pixels LCD. In *Practical Holography X*, volume Proc. SPIE 2652, pages 15–23, Mar. 1996.
- [110] D. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963.
- [111] N. Masuda, T. Ito, T. Tanaka, A. Shiraki, and T. Sugie. Computer generated holography using a graphics processing unit. *Optics Express*, 14(2):603–608, Jan. 2006.
- [112] K. Matsushima. Exact hidden-surface removal in digitally synthetic full-parallax holograms. In *Practical Holography XIX: Materials and Applications*, volume Proc. SPIE 5742, pages 25–32, May 2005.
- [113] K. Matsushima and A. Kondoh. A wave-optical algorithm for hidden-surface removal in digitally synthetic full-parallax holograms for three-dimensional objects. In *Practical Holography XVIII: Materials and Applications*, volume Proc. SPIE 5290, pages 90–97, June 2004.
- [114] K. Matsushima and S. Nakahara. Extremely high-definition full-parallax computer-generated hologram created by the polygon-based method. *Applied Optics*, 48(34):H54–H63, Dec. 2009.
- [115] K. Matsushima, M. Nakamura, and S. Nakahara. Silhouette method for hidden surface removal in computer holography and its acceleration using the switch-back technique. *Optics Express*, 22(20):24450–24465, Oct. 2014.
- [116] K. Matsushima, H. Schimmel, and F. Wyrowski. New creation algorithm for digitally synthesized holograms in surface model by diffraction from tilted planes. In *Practical Holography XVI and Holographic Materials VIII*, volume Proc. SPIE 4659, pages 53–60, June 2002.
- [117] K. Matsushima, H. Schimmel, and F. Wyrowski. Fast calculation method for optical diffraction on tilted planes by use of the angular spectrum of plane waves. *Journal of the Optical Society of America A*, 20(9):1755–1762, Sept. 2003.
- [118] K. Matsushima and M. Takai. Recurrence Formulas for Fast Creation of Synthetic Three-Dimensional Holograms. *Applied Optics*, 39(35):6587–6594, Dec. 2000.
- [119] G. A. Mills and I. Yamaguchi. Effects of quantization in phase-shifting digital holography. *Applied Optics*, 44(7):1216–1225, Mar. 2005.
- [120] T. Mishina, M. Okui, and F. Okano. Calculation of holograms from elemental images captured by integral photography. *Applied Optics*, 45(17):4026–4036, June 2006.

- [121] E. Moon, M. Kim, J. Roh, H. Kim, and J. Hahn. Holographic head-mounted display with RGB light emitting diode light source. *Optics Express*, 22(6):6526–6534, Mar. 2014.
- [122] T. Moons, L. Van Gool, and M. Vergauwen. 3d Reconstruction from Multiple Images Part 1: Principles. *Foundations and Trends in Computer Graphics and Vision*, 4(4):287–404, Apr. 2010.
- [123] K.-D. Na, M.-W. Kwon, S.-C. Kim, and E.-S. Kim. Efficient hologram generation of 3d object using block-based region analysis and N-LUT method. In *Optics and Photonics for Information Processing VI*, volume Proc. SPIE 8498, pages 849815–849815–11, Oct. 2012.
- [124] L. G. Neto, D. Roberge, and Y. Sheng. Full-range, continuous, complex modulation by the use of two coupled-mode liquid-crystal televisions. *Applied Optics*, 35(23):4567–4576, Aug. 1996.
- [125] H. Nishi, K. Matsushima, and S. Nakahara. Rendering of specular surfaces in polygon-based computer-generated holograms. *Applied Optics*, 50(34):H245, Dec. 2011.
- [126] W. Nishii and K. Matsushima. A wavefront printer using phase-only spatial light modulator for producing computer-generated volume holograms. In *Practical Holography XXVIII: Materials and Applications*, volume Proc. SPIE 9006, pages 90061F–90061F–8, Feb. 2014.
- [127] T. Nishitsuji, T. Shimobaba, T. Kakue, N. Masuda, and T. Ito. Fast calculation of computer-generated hologram using the circular symmetry of zone plates. *Optics Express*, 20(25):27496–27502, Dec. 2012.
- [128] Y. Ohsawa and Y. Sakamoto. Computer-generated holograms at arbitrary positions using multi-view images. In *Practical Holography XXVI: Materials and Applications*, volume Proc. SPIE 8281, pages 82810Y–82810Y–8, Feb. 2012.
- [129] Y. Ohsawa, K. Yamaguchi, T. Ichikawa, and Y. Sakamoto. Computer-generated holograms using multiview images captured by a small number of sparsely arranged cameras. *Applied Optics*, 52(1):A167–A176, Jan. 2013.
- [130] N. Okada, T. Shimobaba, Y. Ichihashi, R. Oi, K. Yamamoto, T. Kakue, and T. Ito. Fast calculation of computer-generated hologram for RGB and depth images using wavefront recording plane method. *Photonics Letters of Poland*, 6(3):90–92, Sept. 2014.
- [131] T. Okoshi. *Three-dimensional imaging techniques*. Academic Press, 1976.
- [132] Y. Pan, X. Xu, S. Solanki, X. Liang, R. B. A. Tanjung, C. Tan, and T.-C. Chong. Fast CGH computation using S-LUT on GPU. *Optics Express*, 17(21):18543–18555, Oct. 2009.
- [133] J.-H. Park, N. Chen, G. Baasantseren, M.-Y. Shin, and N. Kim. Hologram generation from orthographic view images of three-dimensional object and its optimization. In *Three-Dimensional Imaging, Visualization, and Display 2009*, volume Proc. SPIE 7329, pages 73290D–73290D–9, May 2009.

- [134] J.-H. Park, M.-S. Kim, G. Baasantseren, and N. Kim. Fresnel and Fourier hologram generation using orthographic projection images. *Optics Express*, 17(8):6320–6334, Apr. 2009.
- [135] J.-H. Park, S.-B. Kim, H.-J. Yeom, H.-J. Kim, H. Zhang, B. Li, Y.-M. Ji, S.-H. Kim, and S.-B. Ko. Continuous shading and its fast update in fully analytic triangular-mesh-based computer generated hologram. *Optics Express*, 23(26):33893, Dec. 2015.
- [136] J.-H. Park, H.-J. Yeom, H.-J. Kim, H. Zhang, B. Li, Y.-M. Ji, and S.-H. Kim. Removal of line artifacts on mesh boundary in computer generated hologram by mesh phase matching. *Optics Express*, 23(6):8006, Mar. 2015.
- [137] K. Patorski. Fraunhofer Diffraction Patterns of Titled Planar Objects. *Optica Acta: International Journal of Optics*, 30(5):673–679, May 1983.
- [138] A.-H. Phan, M. A. Alam, S.-H. Jeon, J.-H. Lee, and N. Kim. Fast hologram generation of long-depth object using multiple wavefront recording planes. In *Practical Holography XXVIII: Materials and Applications*, volume Proc. SPIE 9006, pages 900612–900612–8, Feb. 2014.
- [139] B. T. Phong. Illumination for Computer Generated Pictures. *Communications of the ACM*, 18(6):311–317, June 1975.
- [140] P. Picart and J.-c. Li. *Digital Holography*. John Wiley & Sons, Jan. 2013.
- [141] P. Picart, P. Tankam, D. Mounier, Z.-j. Peng, and J.-c. Li. Spatial bandwidth extended reconstruction for digital color Fresnel holograms. *Optics Express*, 17(11):9145–9156, May 2009.
- [142] W. Plesniak. Incremental update of computer-generated holograms. *Applied Optics*, 42(6):1560–1571, June 2003.
- [143] W. Plesniak, J. Bove, R. Pappu, J. Barabas, and M. Halle. Reconfigurable image projection holograms. *Optical Engineering*, 45(11):115801–115801–15, Nov. 2006.
- [144] H. J. Rabal, N. Bolognini, and E. E. Sicre. Diffraction by a Tilted Aperture. *Optica Acta: International Journal of Optics*, 32(11):1309–1311, Nov. 1985.
- [145] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The Office of the Future: A Unified Approach to Image-based Modeling and Spatially Immersive Displays. In *25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98*, pages 179–188, 1998.
- [146] S. Reichelt, R. Häussler, G. Fütterer, and N. Leister. Depth cues in human visual perception and their realization in 3d displays. In *Three-Dimensional Imaging, Visualization, and Display 2010 and Display Technologies and Applications for Defense, Security, and Avionics IV*, volume Proc. SPIE 7690, pages 76900B–76900B–12, May 2010.
- [147] J. F. Restrepo and J. Garcia-Sucerquia. Magnified reconstruction of digitally recorded holograms by Fresnel-Bluestein transform. *Applied Optics*, 49(33):6430–6435, Nov. 2010.

- [148] Y. Rivenson, A. Stern, and J. Rosen. Compressive Sensing Approach for Reducing the Number of Exposures in Multiple View Projection Holography. In *Frontiers in Optics 2010/Laser Science XXVI*, OSA Technical Digest, page FThM2, Oct. 2010.
- [149] Y. Rivenson, A. Stern, and J. Rosen. Compressive multiple view projection incoherent holography. *Optics Express*, 19(7):6109, Mar. 2011.
- [150] Y. Sakamoto and T. Nagao. A fast computational method for computer-generated Fourier hologram using patch model. *Electronics and Communications in Japan (Part II: Electronics)*, 85(11):16–24, Nov. 2002.
- [151] Y. Sakamoto and A. Tsuruno. A Representation Method for Object Surface Glossiness in Computer-Generated Hologram. *The transactions of the Institute of Electronics, Information and Communication Engineers. D-II*, 88(10):2046–2053, Oct. 2005.
- [152] Y. Sakamoto, Y. Yamashita, and T. Nagao. An Algorithm for Object-Light Calculation Considering Reflectance Distribution for Computer-Generated Holograms. *The Journal of the Institute of Image Information and Television Engineers*, 56(4):611–616, 2002.
- [153] H. Sakata, K. Hosoyachi, C.-Y. Yang, and Y. Sakamoto. Calculation method for computer-generated holograms with cylindrical basic object light by using a graphics processing unit. *Applied Optics*, 50(34):H306–H314, Dec. 2011.
- [154] H. Sakata and Y. Sakamoto. Fast computation method for a Fresnel hologram using three-dimensional affine transformations in real space. *Applied Optics*, 48(34):H212–H221, Dec. 2009.
- [155] H. Sakata and Y. Sakamoto. Pre-calculated object light-based fast calculation method for computer-generated hologram. In *Practical Holography XXIV: Materials and Applications*, volume Proc. SPIE 7619, pages 76190Y–76190Y–9, Feb. 2010.
- [156] D. Scharstein. *View Synthesis Using Stereo Vision*. Springer-Verlag, Berlin, Heidelberg, 1999.
- [157] U. Schnars and W. Jüptner. *Digital Holography: Digital Hologram Recording, Numerical Reconstruction, and Related Techniques*. Springer Science & Business Media, Dec. 2005.
- [158] M. A. Seldowitz, J. P. Allebach, and D. W. Sweeney. Synthesis of digital holograms by direct binary search. *Applied Optics*, 26(14):2788–2798, July 1987.
- [159] N. T. Shaked, B. Katz, and J. Rosen. Fluorescence multicolor hologram recorded by using a macrolens array. *Optics Letters*, 33(13):1461–1463, July 2008.
- [160] N. T. Shaked, B. Katz, and J. Rosen. Review of three-dimensional holographic imaging by multiple-viewpoint-projection based methods. *Applied Optics*, 48(34):H120–H136, Dec. 2009.
- [161] N. T. Shaked and J. Rosen. Modified Fresnel computer-generated hologram directly recorded by multiple-viewpoint projections. *Applied Optics*, 47(19):D21–D27, July 2008.

- [162] N. T. Shaked and J. Rosen. Multiple-viewpoint projection holograms synthesized by spatially incoherent correlation with broadband functions. *Journal of the Optical Society of America A*, 25(8):2129–2138, Aug. 2008.
- [163] N. T. Shaked, J. Rosen, and A. Stern. Integral holography: white-light single-shot hologram acquisition. *Optics Express*, 15(9):5754–5760, Apr. 2007.
- [164] Y. Sheng and J. Wang. Binary hologram of very large space bandwidth product designed by the Genetic Algorithm. In *Digital Holography & 3-D Imaging Meeting*, OSA Technical Digest, page DM4A.3, 2015.
- [165] T. Shimobaba, N. Masuda, and T. Ito. Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane. *Optics Letters*, 34(20):3133–3135, Oct. 2009.
- [166] T. Shimobaba, Y. Nagahama, T. Kakue, N. Takada, N. Okada, Y. Endo, R. Hirayama, D. Hiyama, and T. Ito. Calculation reduction method for color digital holography and computer-generated hologram using color space conversion. *Optical Engineering*, 53(2):024108–024108, Feb. 2014.
- [167] T. Shimobaba, H. Nakayama, N. Masuda, and T. Ito. Rapid calculation algorithm of Fresnel computer-generated-hologram using look-up table and wavefront-recording plane methods for three-dimensional display. *Optics Express*, 18(19):19504–19509, Sept. 2010.
- [168] T. Shimobaba, N. Okada, T. Kakue, N. Masuda, Y. Ichihashi, R. Oi, K. Yamamoto, and T. Ito. Computer Holography using wavefront recording method. In *Digital Holography and Three-Dimensional Imaging (2013)*, OSA Technical Digest, page DTu1A.2, Apr. 2013.
- [169] A. E. Shortt, T. J. Naughton, and B. Javidi. A companding approach for nonuniform quantization of digital holograms of three-dimensional objects. *Optics Express*, 14(12):5129–5134, June 2006.
- [170] A. E. Shortt, T. J. Naughton, and B. Javidi. Compression of digital holograms of three-dimensional objects using wavelets. *Optics Express*, 14(7):2625–2630, Apr. 2006.
- [171] A. E. Shortt, T. J. Naughton, and B. Javidi. Histogram Approaches for Lossy Compression of Digital Holograms of Three-Dimensional Objects. *IEEE Transactions on Image Processing*, 16(6):1548–1556, June 2007.
- [172] S. Silver. Microwave Aperture Antennas and Diffraction Theory\*. *Journal of the Optical Society of America*, 52(2):131–139, Feb. 1962.
- [173] P. Smigielski. *Holographie Optique - Principes*, volume base documentaire : 42528210. Editions T.I., Oct. 1998.
- [174] Q. Y. J. Smithwick, J. Barabas, D. E. Smalley, and J. Bove, V. Michael. Interactive holographic stereograms with accommodation cues. In *Practical Holography XXIV: Materials and Applications*, volume Proc. SPIE 7619, pages 761903–761903–13, Feb. 2010.

- [175] A. Sommerfeld. Mathematische Theorie der Diffraction. *Mathematische Annalen*, 47(2-3):317–374, June 1896.
- [176] A. Symeonidou, D. Blinder, A. Munteanu, and P. Schelkens. Computer-generated holograms by multiple wavefront recording plane method with occlusion culling. *Optics Express*, 23(17):22149, Aug. 2015.
- [177] H. Takahashi, K. Tanaka, H. Okamoto, H. Ueda, and E. Shimizu. Direct volume access by an improved electro-holography image generator. In *Practical Holography IX*, volume Proc. SPIE 2406, pages 220–225, Apr. 1995.
- [178] Y. Takaki and N. Okada. Hologram generation by horizontal scanning of a high-speed spatial light modulator. *Applied Optics*, 48(17):3255–3260, June 2009.
- [179] T. Tommasi and B. Bianco. Frequency analysis of light diffraction between rotated planes. *Optics Letters*, 17(8):556–558, Apr. 1992.
- [180] T. Tommasi and B. Bianco. Computer-generated holograms of tilted planes by a spatial frequency approach. *Journal of the Optical Society of America A*, 10(2):299–305, Feb. 1993.
- [181] K. E. Torrance and E. M. Sparrow. Theory for Off-Specular Reflection From Roughened Surfaces. *Journal of the Optical Society of America*, 57(9):1105, Sept. 1967.
- [182] R. Tudela, I. Labastida, E. Martin-Badosa, S. Vallmitjana, I. Juvells, and A. Carnicer. A simple method for displaying Fresnel holograms on liquid crystal panels. *Optics Communications*, 214(1–6):107–114, Dec. 2002.
- [183] R. Tudela, E. Martin-Badosa, I. Labastida, S. Vallmitjana, and A. Carnicer. Wavefront reconstruction by adding modulation capabilities of two liquid crystal devices. *Optical Engineering*, 43(11):2650–2657, Nov. 2004.
- [184] R. Tudela, E. Martin-Badosa, I. J. I. Labastida, E. Pleguezuelos, S. Vallmitjana, I. P. Juvells, and A. Carnicer. Full complex Fresnel holograms displayed on liquid crystal devices. In *19th Congress of the International Commission for Optics: Optics for the Quality of Life*, volume Proc. SPIE 4829, pages 458–459, Nov. 2003.
- [185] J. S. Underkoffler. Occlusion processing and smooth surface shading for fully computed synthetic holography. In *Practical Holography XI and Holographic Materials III*, volume Proc. SPIE 3011, pages 19–30, Apr. 1997.
- [186] N. Verrier and M. Atlan. Off-axis digital hologram reconstruction: some practical considerations. *Applied Optics*, 50(34):H136–H146, Dec. 2011.
- [187] K. Viswanathan, P. Gioia, and L. Morin. Wavelet compression of digital holograms: Towards a view-dependent framework. In *Applications of Digital Image Processing XXXVI*, volume Proc. SPIE 8856, pages 88561N–88561N–10, Sept. 2013.
- [188] K. Viswanathan, P. Gioia, and L. Morin. Morlet Wavelet transformed holograms for numerical adaptive view-based reconstruction. In *Optics and Photonics for Information Processing VIII*, volume Proc. SPIE 9216, pages 92160G–92160G–14, Sept. 2014.

- [189] K. Wakunami and M. Yamaguchi. Calculation for computer generated hologram using ray-sampling plane. *Optics Express*, 19(10):9086–9101, May 2011.
- [190] K. Wakunami and M. Yamaguchi. Occlusion processing for computer generated hologram by conversion between the wavefront and light-ray information. *Journal of Physics: Conference Series*, 415(1):012047, Feb. 2013.
- [191] K. Wakunami, H. Yamashita, and M. Yamaguchi. Occlusion culling for computer generated hologram based on ray-wavefront conversion. *Optics Express*, 21(19):21811–21822, Sept. 2013.
- [192] J. Weng, T. Shimobaba, N. Okada, H. Nakayama, M. Oikawa, N. Masuda, and T. Ito. Generation of real-time large computer generated hologram using wavefront recording method. *Optics Express*, 20(4):4018–4023, Feb. 2012.
- [193] C. Wheatstone. Contributions to the Physiology of Vision.–Part the First. On Some Remarkable, and Hitherto Unobserved, Phenomena of Binocular Vision. *Philosophical Transactions of the Royal Society of London*, 128:371–394, Jan. 1838.
- [194] Y. Xing, M. Kaaniche, B. Pesquet-Popescu, and F. Dufaux. Vector lifting scheme for phase-shifting holographic data compression. *Optical Engineering*, 53(11):112312–112312, May 2014.
- [195] Y. Xing, M. Kaaniche, B. Pesquet-Popescu, and F. Dufaux. Adaptive nonseparable vector lifting scheme for digital holographic data compression. *Applied Optics*, 54(1):A98–A109, Jan. 2015.
- [196] Y. Xing, M. Kaaniche, B. Pesquet-Popescu, and F. Dufaux. *Digital Holographic Data Representation and Compression*. Academic Press, Oct. 2015.
- [197] Y. Xing, B. Pesquet-Popescu, and F. Dufaux. Compression of computer generated phase-shifting hologram sequence using AVC and HEVC. In *Applications of Digital Image Processing XXXVI*, volume Proc. SPIE 8856, pages 88561M–88561M–8, Sept. 2013.
- [198] Y. Xing, B. Pesquet-Popescu, and F. Dufaux. Comparative study of scalar and vector quantization on different phase-shifting digital holographic data representations. In *2014 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4, July 2014.
- [199] K. Yamaguchi, T. Ichikawa, and Y. Sakamoto. Calculation method for computer-generated holograms considering various reflectance distributions based on microfacets with various surface roughnesses. *Applied Optics*, 50(34):H195–H202, Dec. 2011.
- [200] K. Yamaguchi and Y. Sakamoto. Computer generated hologram with characteristics of reflection: reflectance distributions and reflected images. *Applied Optics*, 48(34):H203–H211, Dec. 2009.
- [201] K. Yamaguchi and Y. Sakamoto. Computer-generated holograms considering background reflection on various object shapes with reflectance distributions. In *Practical Holography XXIV: Materials and Applications*, volume Proc. SPIE 7619, pages 761909–761909–11, Feb. 2010.

- [202] M. Yamaguchi. Image-based computational holography for deep 3d scene display. In *2013 International Conference on Optical Instruments and Technology: Optical Systems and Modern Optoelectronic Instruments*, volume Proc. SPIE 9042, pages 90420A–90420A–5, Dec. 2013.
- [203] M. Yamaguchi, H. Hoshino, T. Honda, and N. Ohyama. Phase-added stereogram: calculation of hologram using computer graphics technique. In *Practical Holography VII: Imaging and Materials*, volume Proc. SPIE 1914, pages 25–31, Sept. 1993.
- [204] M. Yamaguchi, K. Wakunami, and M. Inaniwa. Computer generated hologram from full-parallax 3d image data captured by scanning vertical camera array. *Chinese Optics Letters*, 12(6):060018, June 2014.
- [205] T. Yamaguchi, G. Okabe, and H. Yoshikawa. Real-time image plane full-color and full-parallax holographic video display system. *Optical Engineering*, 46(12):125801–125801–8, Dec. 2007.
- [206] T. Yamaguchi and H. Yoshikawa. Computer-generated image hologram. *Chinese Optics Letters*, 9(12):120006, Dec. 2011.
- [207] K. Yamamoto, T. Mishina, R. Oi, T. Senoh, and T. Kurita. Real-time color holography system for live scene using 4k2k video system. In *Practical Holography XXIV: Materials and Applications*, volume Proc. SPIE 7619, pages 761906–761906–10, Feb. 2010.
- [208] K. Yamamoto, T. Senoh, R. Oi, and T. Kurita. 8k4k-size computer generated hologram for 3-D visual system using rendering technology. In *Universal Communication Symposium (IUCS), 2010 4th International*, pages 193–196, Oct. 2010.
- [209] F. Yaras, H. Kang, and L. Onural. State of the Art in Holographic Displays: A Survey. *Journal of Display Technology*, 6(10):443–454, Oct. 2010.
- [210] T. Yatagai. Stereoscopic approach to 3-D display using computer-generated holograms. *Applied Optics*, 15(11):2722–2729, Nov. 1976.
- [211] K. Yee. Numerical solution of initial boundary value problems involving maxwell’s equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 14(3):302–307, May 1966.
- [212] H.-J. Yeom, H.-J. Kim, S.-B. Kim, H. Zhang, B. Li, Y.-M. Ji, S.-H. Kim, and J.-H. Park. 3d holographic head mounted display using holographic optical elements with astigmatism aberration compensation. *Optics Express*, 23(25):32025–32034, Dec. 2015.
- [213] H. Yoshikawa, S. Iwase, and T. Oneda. Fast computation of Fresnel holograms employing difference. In *Practical Holography XIV and Holographic Materials VI*, volume Proc. SPIE 3956, pages 48–55, Mar. 2000.
- [214] H. Yoshikawa and H. Kameyama. Integral holography. In *Practical Holography IX*, volume Proc. SPIE 2406, pages 226–234, Apr. 1995.

- [215] H. Yoshikawa and T. Yamaguchi. Review of Holographic Printers for Computer-Generated Holograms. *IEEE Transactions on Industrial Informatics*, PP(99):1–1, Sept. 2015.
- [216] H. Yoshikawa, T. Yamaguchi, and R. Kitayama. Real-Time Generation of Full Color Image Hologram with Compact Distance Look-up Table. In *Advances in Imaging*, OSA Technical Digest, page DWC4, Apr. 2009.
- [217] N. Yoshikawa, M. Itoh, and T. Yatagai. Use of genetic algorithm for computer-generated holograms. In *International Conference on Applications of Optical Holography*, volume Proc. SPIE 2577, pages 150–157, July 1995.
- [218] N. Yoshikawa and T. Yatagai. Phase optimization of a kinoform by simulated annealing. *Applied Optics*, 33(5):863–868, Feb. 1994.
- [219] L. Yu, Y. An, and L. Cai. Numerical reconstruction of digital holograms with variable viewing angles. *Optics express*, 10(22):1250–1257, Nov. 2002.
- [220] L. Yu, Y. An, and L. Cai. Numerical reconstruction of digital holography with changed viewing angles. *Journal of Electronic Imaging*, 13(4):814–818, Oct. 2004.
- [221] H. Zhang, N. Collings, J. Chen, B. Crossland, D. Chu, and J. Xie. Full parallax three-dimensional display with occlusion effect using computer generated hologram. *Optical Engineering*, 50(7):074003–074003–5, July 2011.
- [222] H. Zhang, Q. Tan, and G. Jin. Full parallax three-dimensional computer generated hologram with occlusion effect using ray casting technique. *Journal of Physics: Conference Series*, 415(1):012048, Feb. 2013.
- [223] H. Zhang, Y. Zhao, L. Cao, and G. Jin. Three dimensional computer-generated holograms with shading effects based on a phase-only spatial light modulator. In *International Symposium on Photoelectronic Detection and Imaging 2013: Optical Storage and Display Technology*, volume Proc. SPIE 8913, pages 891310–891310–6, Aug. 2013.
- [224] H. Zhang, Y. Zhao, L. Cao, and G. Jin. Calculation for computer-generated holograms using fully computed holographic stereogram based method. In *Digital Holography & 3-D Imaging Meeting*, OSA Technical Digest, page DM4A.4, 2015.
- [225] H. Zhang, Y. Zhao, L. Cao, and G. Jin. Layered holographic stereogram based on inverse Fresnel diffraction. *Applied Optics*, 55(3):A154, Jan. 2016.
- [226] L. Zhang and W. J. Tam. Stereoscopic image generation based on depth images for 3d TV. *IEEE Transactions on Broadcasting*, 51(2):191–199, June 2005.
- [227] Y. Zhao, L. Cao, H. Zhang, D. Kong, and G. Jin. Accurate calculation of computer-generated holograms using angular-spectrum layer-oriented method. *Optics Express*, 23(20):25440, Oct. 2015.
- [228] Y. Zhao, L. Cao, H. Zhang, W. Tan, S. Wu, Z. Wang, Q. Yang, and G. Jin. Time-division multiplexing holographic display using angular-spectrum layer-oriented method (Invited Paper). *Chinese Optics Letters*, 14(1):010005, Jan. 2016.

- [229] J.-Y. Zhuang and O. K. Ersoy. Fast decimation-in-frequency direct binary search algorithms for synthesis of computer-generated holograms. *JOSA A*, 11(1):135–143, Jan. 1994.
- [230] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality Video View Interpolation Using a Layered Representation. In *31st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '04, pages 600–608, 2004.

# Appendix A

## Publications and patent

### A.1 Scientific journals

[J1] **Antonin Gilles**, Patrick Gioia, Rémi Cozot and Luce Morin, "Génération Numérique d'Hologrammes : Etat de l'Art", in *Revue Electronique Francophone d'Informatique Graphique (REFIG)*, Vol. 9, No 1, pages 23-35, 2015.

*Abstract* – This paper reviews Computer Generated Holography techniques applied to 3D video. Using these methods, it is possible to acquire holograms of synthetic or existing scenes without physical interference between light waves.

[J2] **Antonin Gilles**, Patrick Gioia, Rémi Cozot and Luce Morin, "Hybrid approach for fast occlusion processing in computer generated hologram calculation", in *Applied Optics, The Optical Society*, Vol. 55, No 20, pages 5459-5470, 2016.

*Abstract* – A hybrid approach for fast occlusion processing in computer generated hologram calculation is studied in this paper. The proposed method is based on the combination of two commonly used approaches, which complement one another: the point-source and wave-field approaches. By using these two approaches together, the proposed method thus takes advantages from both of them. In this method, the 3D scene is first sliced into several depth layers parallel to the hologram plane. Light scattered by the scene is then propagated and shielded from one layer to another using either a point-source or a wave-field approach according to a threshold criterion on the number of points within the layer. Finally, the hologram is obtained by computing the propagation of light from the nearest layer to the hologram plane. Experimental results reveal that the proposed method does not produce any visible artifact and outperforms both the point-source and wave-field approaches.

## A.2 International conferences

[C1] **Antonin Gilles**, Patrick Gioia, Rémi Cozot and Luce Morin, "Complex modulation computer-generated hologram by a fast hybrid point-source/wave-field approach", in *IEEE International Conference on Image Processing (ICIP)*, Quebec City, Canada, 2015.

*Abstract* – We propose a fast Computer-Generated Hologram (CGH) computation method based on a hybrid point-source/wave-field approach. Whereas previously proposed methods tried to reduce the computational complexity of the point-source or the wave-field approaches independently, our method uses the two approaches together and therefore takes advantages from both of them. The algorithm consists of three steps. First, the 3D scene is sliced into several depth layers parallel to the hologram plane. Then, for each layer, we compute the complex wave scattered by this layer either using a wave-field or a point-source approach according to a threshold criterion on the number of points within the layer. Finally, we sum up the complex waves scattered by all the depth layers to obtain the CGH. Experimental results reveal that this combination of approaches does not produce any visible artifact and outperforms both the point-source and wave-field approaches.

[C2] **Antonin Gilles**, Patrick Gioia, Rémi Cozot and Luce Morin, "Fast generation of complex modulation video holograms using temporal redundancy compression and hybrid point-source/wave-field approaches", in *SPIE Optics + Photonics*, San Diego, USA, 2015.

*Abstract* – The hybrid point-source/wave-field method is a newly proposed approach for Computer-Generated Hologram (CGH) calculation, based on the slicing of the scene into several depth layers parallel to the hologram plane. The complex wave scattered by each depth layer is then computed using either a wave-field or a point-source approach according to a threshold criterion on the number of points within the layer. Finally, the complex waves scattered by all the depth layers are summed up to obtain the CGH. Although outperforming both point-source and wave-field methods without producing any visible artifact, this approach has not yet been used for animated holograms, and the possible exploitation of temporal redundancies has not been studied. In this paper, we propose a fast computation of video holograms by taking into account those redundancies. Our algorithm consists of three steps. First, intensity and depth data of the current 3D video frame are extracted and compared with those of the previous frame to remove temporally redundant data. Then the CGH pattern for this compressed frame is generated using the hybrid point-source/wave-field approach. The resulting CGH pattern is finally transmitted to the video output and stored in the previous frame buffer. Experimental results reveal that our proposed method is able to produce video holograms at interactive rates without producing any visible artifact.

[C3] **Antonin Gilles**, Patrick Gioia, Rémi Cozot and Luce Morin, "Complex modulation computer-generated hologram with occlusion effect by a fast hybrid point-source/wave-field approach", in *Pacific Graphics*, Beijing, China, 2015.

*Abstract* – We propose a fast Computer-Generated Hologram (CGH) computation method with occlusion effect based on a hybrid point-source/wave-field approach. Whereas previously proposed methods tried to reduce the computational complexity of the point-source or the wave-field approaches independently, the proposed method uses the two approaches together and therefore takes advantages from both of them. Our algorithm consists of three steps. First, the 3D scene is sliced into several depth layers parallel to the hologram plane. Then, light scattered by the scene is propagated and shielded from one layer to another, starting from the farthest layer. For each layer, light propagation and light shielding are performed using either a point-source or a wave-field approach according to a threshold criterion on the number of points within the layer. Finally, we compute light propagation from the nearest layer to the hologram plane to obtain the CGH. Experimental results reveal that this combination of approaches does not produce any visible artifact and outperforms both the point-source and wavefield approaches.

[C4] **Antonin Gilles**, Patrick Gioia, Rémi Cozot and Luce Morin, "Computer-generated hologram from Multiview-plus-Depth data considering specular reflections", in *IEEE International Workshop on Hot Topics in 3D (Hot3D)*, Seattle, USA, 2016.

*Abstract* – A novel approach for hologram computation from Multiview-plus-Depth (MVD) data is studied in this paper. The proposed method consists of three steps. First, intensity views and associated depth maps of the scene are taken from different perspective viewpoints. Then, the 3D scene geometry is reconstructed from the MVD data as a layered point-cloud. This 3D scene reconstruction step allows us to use only a few perspective projections of the scene without sacrificing any depth cue. Furthermore, to take into account specular reflections, each scene point is considered to emit light differently in all the directions. Finally, light scattered by the scene is numerically propagated towards the hologram plane to get the CGH. Experimental results show that the proposed method is able to provide all the human depth cues and accurate shading of the scene without producing any visible artifact.

## A.3 Patent

[P1] **Antonin Gilles**, Patrick Gioia, Rémi Cozot and Luce Morin, "Procédé de génération numérique d'un hologramme, dispositif et programme d'ordinateur associés", France, FR15/54363, 2015.



## AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

**Titre de la thèse:**

Fast hologram synthesis methods for realistic 3D visualization

**Nom Prénom de l'auteur : GILLES ANTONIN**

**Membres du jury :**

- Madame GUILLEMOT Christine
- Monsieur DUFAUX Frédéric
- Madame MORIN Luce
- Monsieur COZOT Rémi
- Monsieur GIOIA Patrick
- Monsieur Lucas Laurent
- Monsieur PICART Pascal
- Monsieur SCHELKENS Peter

Président du jury : *Pascal PICART*

Date de la soutenance : 13 Septembre 2016

Reproduction de la these soutenue

- Thèse pouvant être reproduite en l'état  
 ~~Thèse pouvant être reproduite après corrections suggérées~~

Fait à Rennes, le 13 Septembre 2016

Signature du président de jury

Le Directeur,

M'hamed DRISSI



*Pascal PICART*

A stylized handwritten signature in black ink.





L'holographie est souvent considérée comme la technologie de visualisation 3D la plus prometteuse, puisqu'elle fournit l'illusion du relief la plus naturelle et la plus réaliste possible. Toutefois, pour trouver application dans le domaine de la visioconférence ou de la téléprésence, les méthodes de génération numérique d'hologrammes doivent produire des scènes réalistes avec une forte illusion de profondeur, et ce en temps réel. Cette thèse se situe dans ce contexte et est organisée en deux parties.

Dans la première partie de ce travail, nous avons proposé deux algorithmes de synthèse d'hologrammes permettant de se rapprocher du temps réel. Tout d'abord, nous avons développé une méthode combinant deux approches complémentaires: les approches *point-source* et *wave-field*. Alors que les méthodes de l'état de l'art réduisent la complexité de calcul de ces deux approches indépendamment, notre méthode tire parti de chacune d'entre elles. De cette manière, le temps de calcul de l'hologramme a été réduit de plus de 65% par rapport aux méthodes *point-source* et *wave-field* conventionnelles.

Deuxièmement, nous avons cherché à accélérer cette méthode hybride en supprimant les redondances temporelles entre les trames consécutives d'une vidéo 3D. Pour chaque image, l'algorithme détecte les changements dans la scène et met à jour l'hologramme des points affectés. Étant donné que seules de petites régions de l'hologramme sont mises à jour, la charge de calcul est considérablement réduite, permettant le calcul d'hologrammes couleur à 60 images par seconde.

Dans la deuxième partie de ce travail, nous avons proposé deux algorithmes afin d'améliorer la qualité visuelle des scènes. Tout d'abord, nous avons perfectionné la méthode hybride pour tenir compte des occultations dans la scène. Pour cela, nous avons conçu une méthode efficace pour le calcul de l'occultation d'une onde lumineuse par un point. Cette méthode reproduit les occultations sans augmenter de manière significative le temps de calcul de la méthode hybride originale.

Enfin, nous avons proposé une méthode pour le calcul d'un hologramme à partir de données multivue-plus-profondeur (MVD) avec prise en compte des réflexions spéculaires. Selon cette méthode, la géométrie de la scène est reconstruite à partir des données MVD sous la forme d'un nuage de points, ce qui permet de n'utiliser que quelques projections de la scène. En outre, afin de tenir compte des réflexions spéculaires, chaque point de la scène est considéré avoir une émission différente selon les directions. Enfin, la lumière émise par la scène est propagée dans le plan de l'hologramme. Les résultats expérimentaux montrent que cette méthode reproduit tous les indices de la profondeur et l'illumination précise de la scène avec une complexité de calcul réduite.

Holography is often considered as the most promising 3D visualization technology, since it can produce the most realistic and natural depth illusion to the naked eye. However, in order to have application in the field of videoconferencing or telepresence systems, hologram synthesis methods should be able to produce realistic 3D scenes with strong depth illusion in real-time. This thesis falls within this context and is organized into two parts.

In the first part of this work, we investigated two novel algorithms in order to get closer to real-time computation. First, we designed a fast hologram calculation method by combining two approaches which complement one another: the point-source and wave-field approaches. Whereas previously proposed methods reduced the computational complexity of these approaches independently, our method takes advantages from both of them. By this way, the hologram calculation time has been reduced by more than 65% compared to the conventional point-source and wave-field methods.

Second, we further accelerated this hybrid method by removing temporal redundancies between consecutive frames of a 3D video. For each video frame, the algorithm detects changes in the scene and updates the hologram of only affected scene points. Since only small regions of the hologram are updated at each video frame, this method allows the computational burden to be dramatically reduced, enabling the computation of colorful video holograms at 60 frames per second.

In the second part of this work, we proposed two algorithms in order to enhance the visual quality of displayed scenes. First, we improved the hybrid method to take into account occlusions between objects in the scene. To this end, we designed an efficient algorithm for light shielding between points and light waves. Experimental results revealed that this method provides occlusion effect without significantly increasing the hologram calculation time of the original hybrid method.

Finally, we proposed a hologram computation method from Multiview-plus-depth (MVD) data with rendering of specular reflections. In this method, the 3D scene geometry is first reconstructed from the MVD data as a layered point-cloud, enabling the use of only a few perspective projections of the scene. Furthermore, in order to take into account specular reflections, each scene point is considered to emit light differently in all the directions. Finally, light scattered by the scene is numerically propagated towards the hologram plane in order to get the final hologram. Experimental results show that the proposed method is able to provide all the human depth cues and accurate shading of the scene with reduced computational complexity.