

# IFT 6145

## Vision tridimensionnelle

### Environnement immersifs

Sébastien Roy  
Département d'informatique et de recherche opérationnelle  
Université de Montréal  
hiver 2004

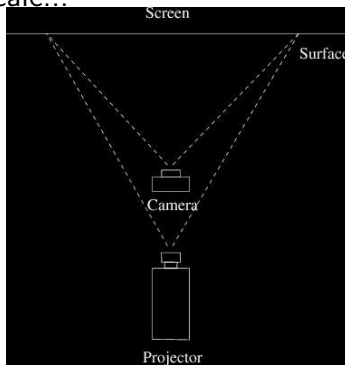
### Au programme

- Projection sur grand écran et immersion
- Le problème de l'alignement de projecteur
- La perspective de l'observateur : la caméra
- Correspondance caméra - projecteur
- Lumière structurée
- Projection distordionnée
- Applications

1

### Projection sur grand écran et immersion

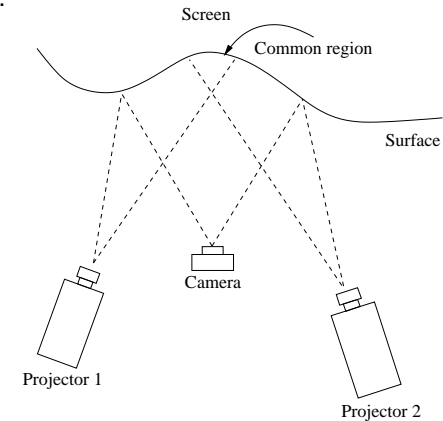
La situation idéale...



2

### Projection sur grand écran et immersion

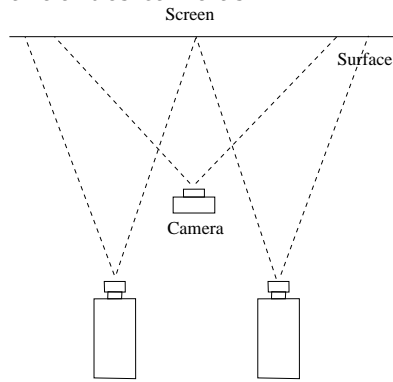
La réalité...



3

## Le problème de l'alignement de projecteur

Alignement manuel des caméras...



⇒ Projecteurs calibrés, Écran plat, Axes optiques parallèles

4

## La perspective de l'observateur : la caméra

On peut remplacer l'observateur par une caméra.

⇒ si c'est beau pour la caméra, c'est beau pour l'observateur

### Si on connaît

- ... quel point du projecteur correspond à un point de la caméra

### alors

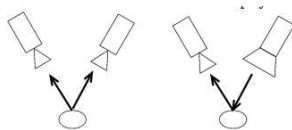
- ... on peut construire une image *projecteur* pour obtenir une image *caméra*.

5

## Correspondance caméra - projecteur

Comment savoir quel pixel du projecteur correspond à un pixel de la caméra ?

⇒ Problème classique de la **Mise en correspondance**



### 2 caméras

Stéréoscopie

### 1 caméra + 1 projecteur

Lumière structurée

6

## Reconstruction 3D active et passive

### Reconstruction passive

- Aucune interaction avec l'environnement
- Utilise uniquement des caméras
- Exemple : mise en correspondance stéréo, flux optique

### Reconstruction active

- On peut *modifier* l'environnement pour simplifier la reconstruction
- Typiquement, on contrôle l'éclairage (laser, projecteur, etc...)
- On peut aussi utiliser les ombres (voir *Weakly structured lighting*, Bouget & Perona)
- Exemple : scanner au laser, ...
- Très utilisé dans l'industrie (95% des scanners)

7

## Reconstruction active et passive

### Problèmes avec le stéréo passif

- Dépendance à la texture des images (zones lisses)
- Ambiguïté de mise correspondance (intensités similaires)
- Spécularités
- Oclusions

### Solution 1 : Stéréoscopie *texturisée*

- On projète une lumière contenant une texture sur la scène
- Mise en correspondance habituelle avec deux caméras
- → Plus de problème de zone de texture lisse !

### Solution 2 : Stéréoscopie à lumière structurée

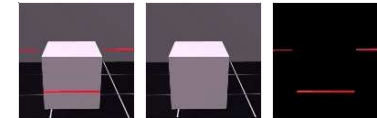
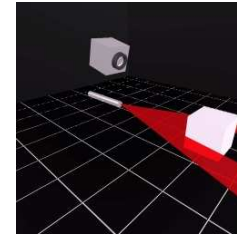
- Transforme une des deux caméras en projecteur
- Élimine le problème de la mise en correspondance

8

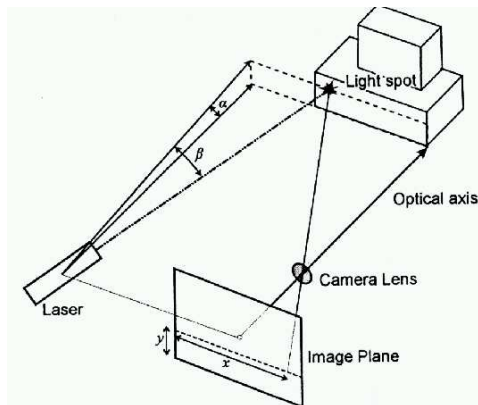
## Lumière structurée

### Projecteur

- laser (point, ligne, etc...)
- projecteur diapo, LCP, DLP, ...



9



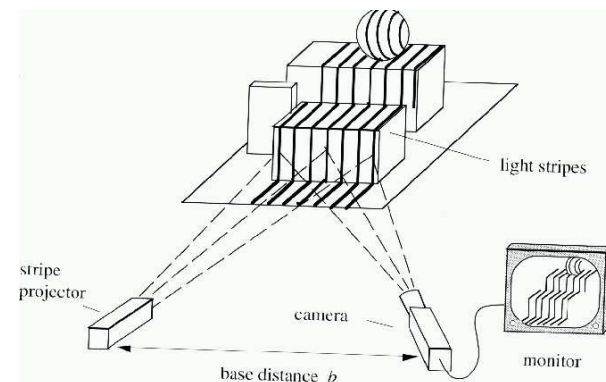
Il suffit de trouver le point laser dans l'image...

10

## Pattern de lumière

Plusieurs lignes à la fois...

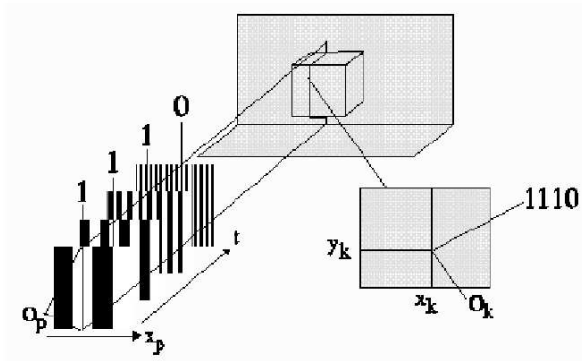
Utiliser un projecteur plutôt qu'un laser !



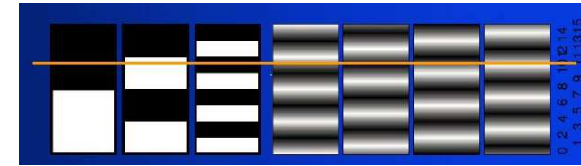
11

## Pattern de lumière

Pour mettre en correspondance, il suffit d'identifier chaque pixel du projecteur. En projetant des images successives choisies, chaque pixel est associé à un *code* qui donne sa position dans l'image.



12



Comment choisir le code ?

13

## Gray Code

- Inventé par Frank Gray en 1953
- Représente une séquence de nombres de  $0$  à  $2^N - 1$  par une séquence de codes
- Chaque code dans la séquence ne diffère de son voisin que d'un bit.
- Pour les nombres  $0 \dots 7$ , nous avons la séquence 000, 001, 011, 010, 110, 111, 101, 100.
- Plusieurs séquences satisfont la propriété d'adjacence.

Comment transformer un nombre en code de Gray et vice-versa ?

14

## Gray Code

- Soit un nombre  $N$  représenté par un tableau de  $n$  bits  
Utiliser l'index  $0$  pour le bit le moins significatif  
Par exemple  $N = 12 \rightarrow N[3] = 1, N[2] = 1, N[1] = 0, N[0] = 0$ .
- Mettre  $0$  dans  $N[n]$
- Le code de gray  $G$  représenté par un tableau de  $n$  bits est

$$G[i] = \text{XOR}(N[i + 1], N[i]) \quad i \in [n - 1, \dots, 0]$$

- L'inverse s'obtient comme suit :

$$N[n - 1] = G[n - 1]$$

$$N[i] = \text{XOR}(N[i + 1], G[i]) \quad i \in [n - 2, \dots, 0]$$

15

## Reconstruction

### Pourquoi le code de Gray ?

Le code d'un pixel ne peut différer de son voisin que d'un seul bit.

Si il diffère de plus, c'est une erreur.

### Pour reconstruire

- Envoyer les codes avec le projecteur
- Calculer les codes de tous les pixels de la caméra.

Et voilà ! La correspondance est établie.

16

## Le point de vue du projecteur

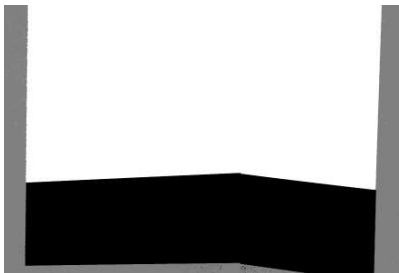
Le projecteur envoie le code...



17

## Le point de vue de la caméra

La caméra regarde le code...



18

## Lumière structurée



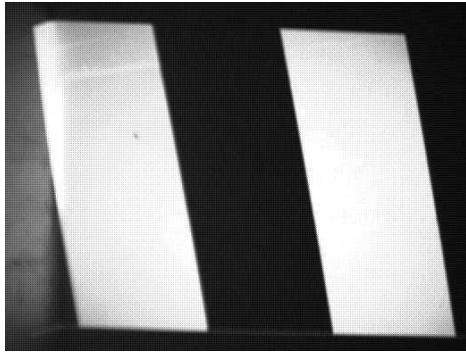
On encode les positions en X et Y.

⇒ Positions inconnues de la caméra et du projecteur

19

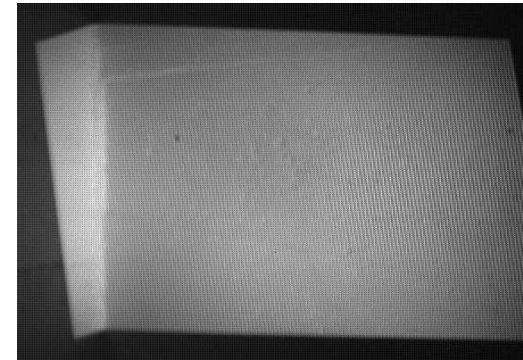
## Précision des code

- Résolution caméra VS résolution projecteur
- Distance avec l'écran



20

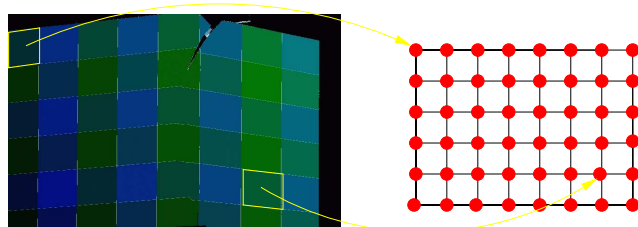
## Bits de poids faible



Souvent le blanc-noir est du gris-gris...

21

## Problème de résolution



Camera image

Projector correspondance

Un point de la caméra peut correspondre à plusieurs points du projecteur.

22

## La méthode du bit masqué

Chaque bit du code peut avoir une valeur **0**, **1**, et **X**.

Projecteur : **0 1 1 0 0 1 1**

Perçu par la caméra : **0 1 1 0 X X X**

**Que faire ?**

On demande un peu d'aide aux pixels voisins...

Si les voisins sont

**0 1 1 0 0 X X**

**0 1 1 0 X 1 X**

**0 1 1 0 X 1 1**

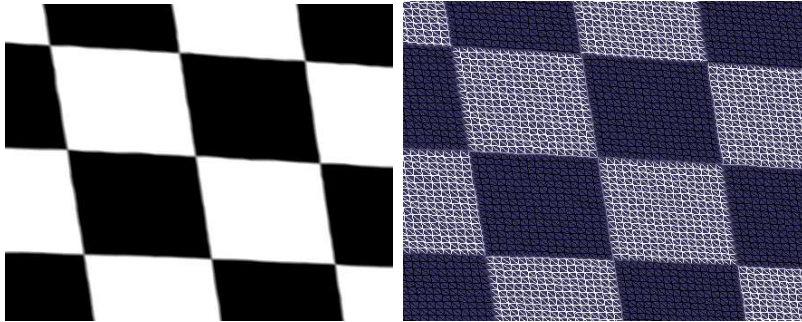
On peut retrouver les bits qui manquent....

23

## Projection distordionnée

Comment projeter...

OpenGL peut afficher une grille et une texture...



24

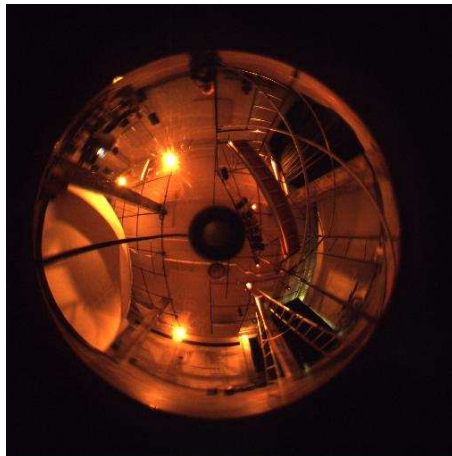
## Élimination de la distorsion



25

## Pas de calibration ?

Puisque on distordionne pour le point de vue de la caméra, aucune calibration explicite n'est requise.



26

## Projecteur multiples

Chaque projecteur donne une reconstruction.

Toutes les reconstructions donnent une grille OpenGL.

On peut projeter simultanément à travers tous les projecteurs.

27

## Le futur

- Caméras multiples
- Contrôle de l'intensité
- Détection des ombres
- Balance automatique des couleurs
- ...