IFT 6145
Vision tridimensionelle

Environnement immersifs

Sébastien Roy
Département d'informatique et de recherche opérationelle
Université de Montréal

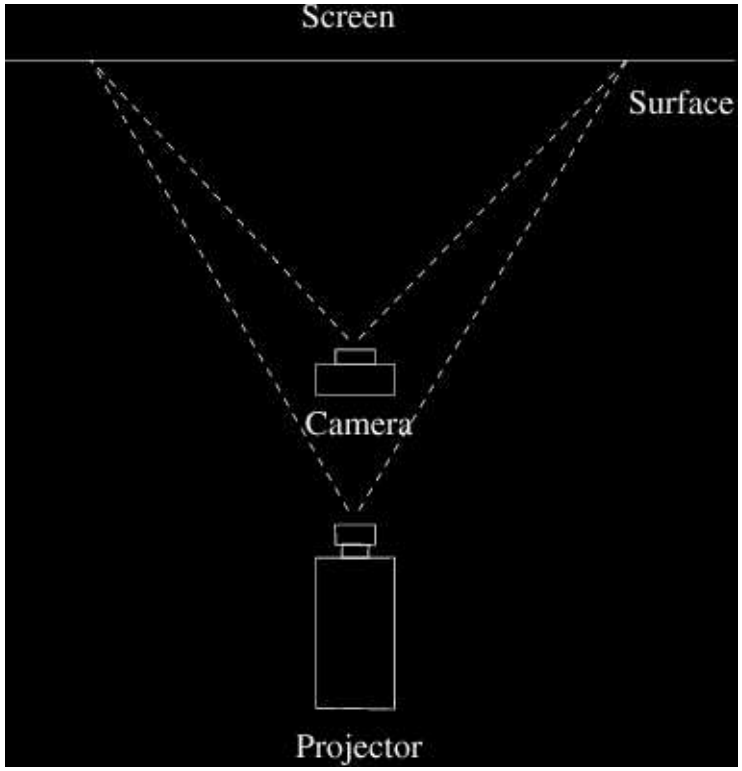
hiver 2004

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 distortionnée
- Applications

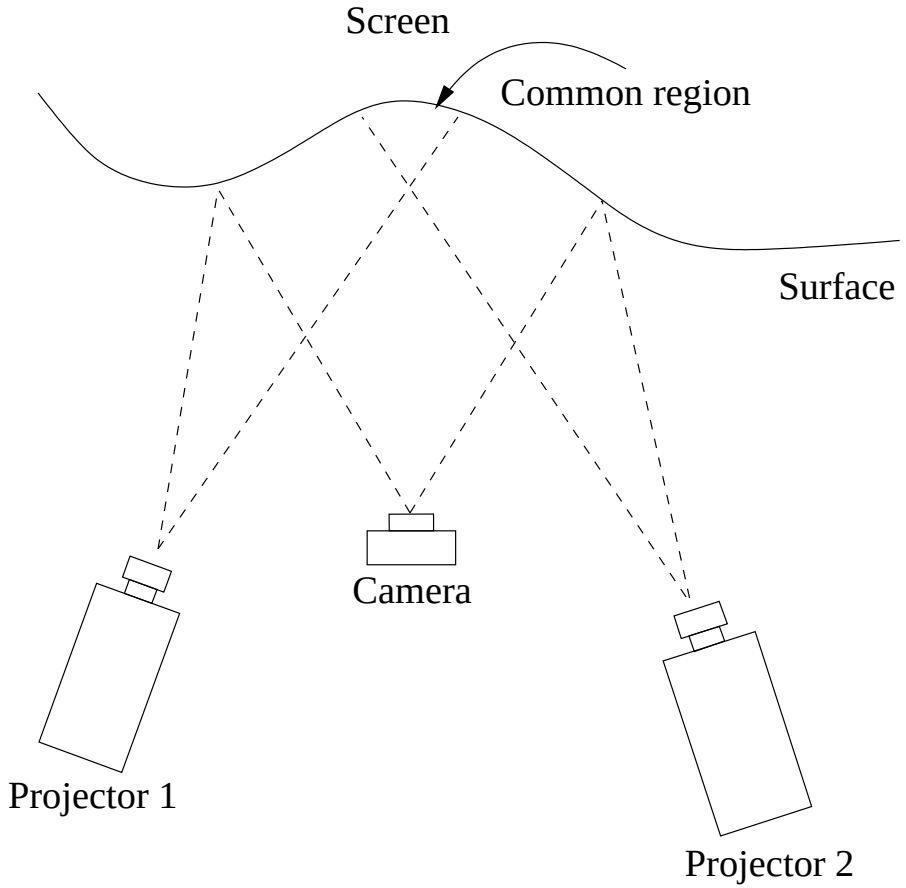
Projection sur grand écran et immersion

La situation idéale...



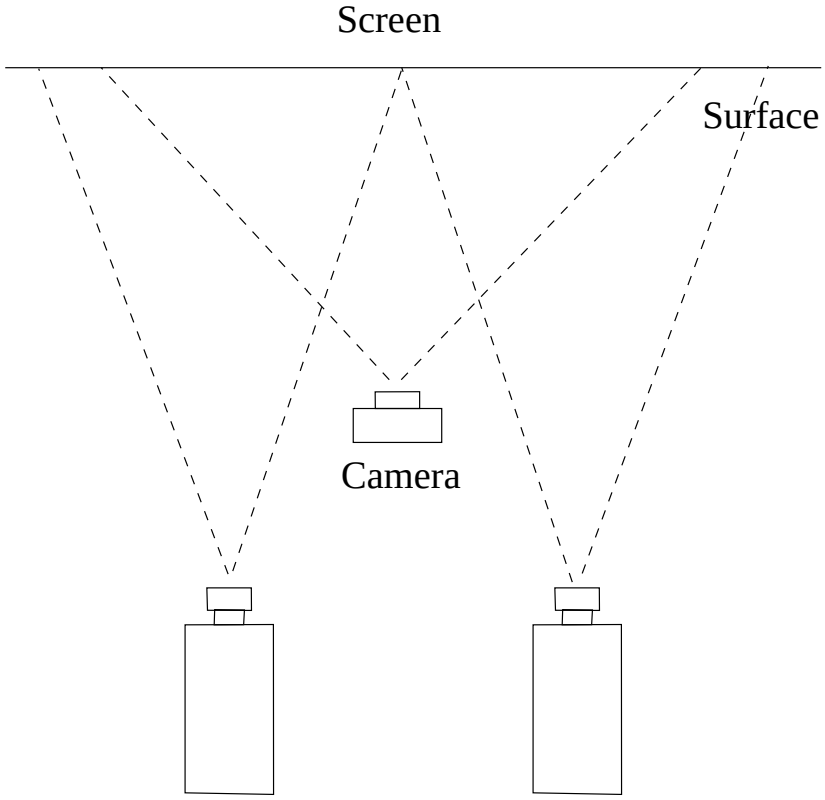
Projection sur grand écran et immersion

La réalité...



Le problème de l'alignement de projecteur

Alignement manuel des caméras...



$\Rightarrow$ Projecteurs calibrés, Écran plat, Axes optiques parallèles

La perspective de l'observateur : la caméra

On peut remplacer l'observateur par une caméra.

$\Rightarrow$ si c'est beau pour la caméra, c'est beau pour l'observateur

Si on connait

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

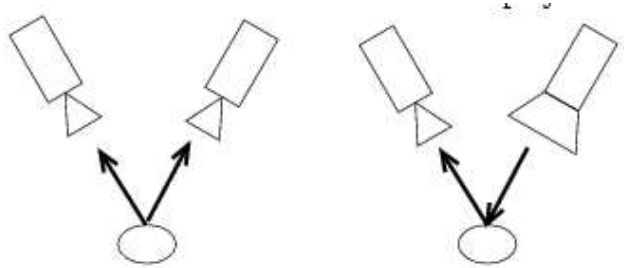
alors

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

Correspondance caméra - projecteur

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

$\Rightarrow$ Problème classique de le **Mise en correspondance**



2 caméras

Stéréoscopie

1 caméra + 1 projecteur

Lumière structurée

Reconstruction 3D active et passive

Reconstruction passive

- Aucune intéraction 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)

Reconstruction active et passive

Problèmes avec le stéréo passif

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

Solution 1 : Stéréoscopie *texturisée*

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

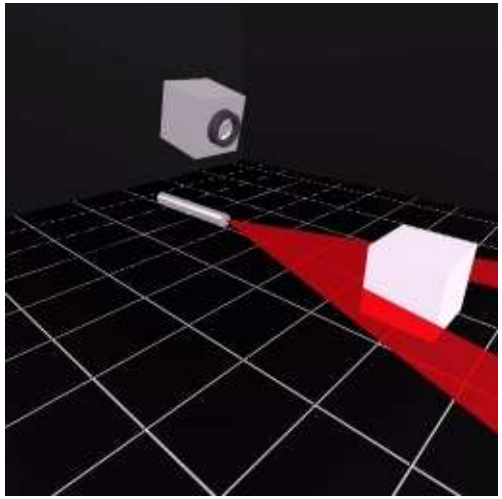
Solution 2 : Stéréoscopie à lumière structurée

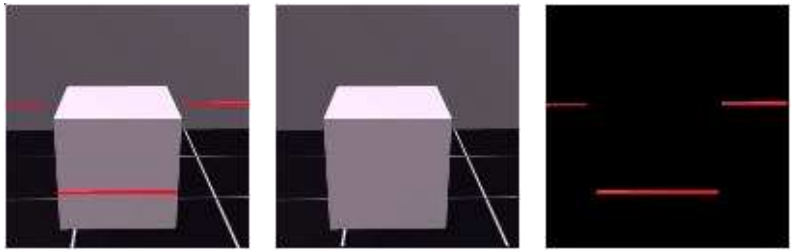
- Transforme une des deux caméras en projecteur
- Élimine le problème de la mise en correspondance

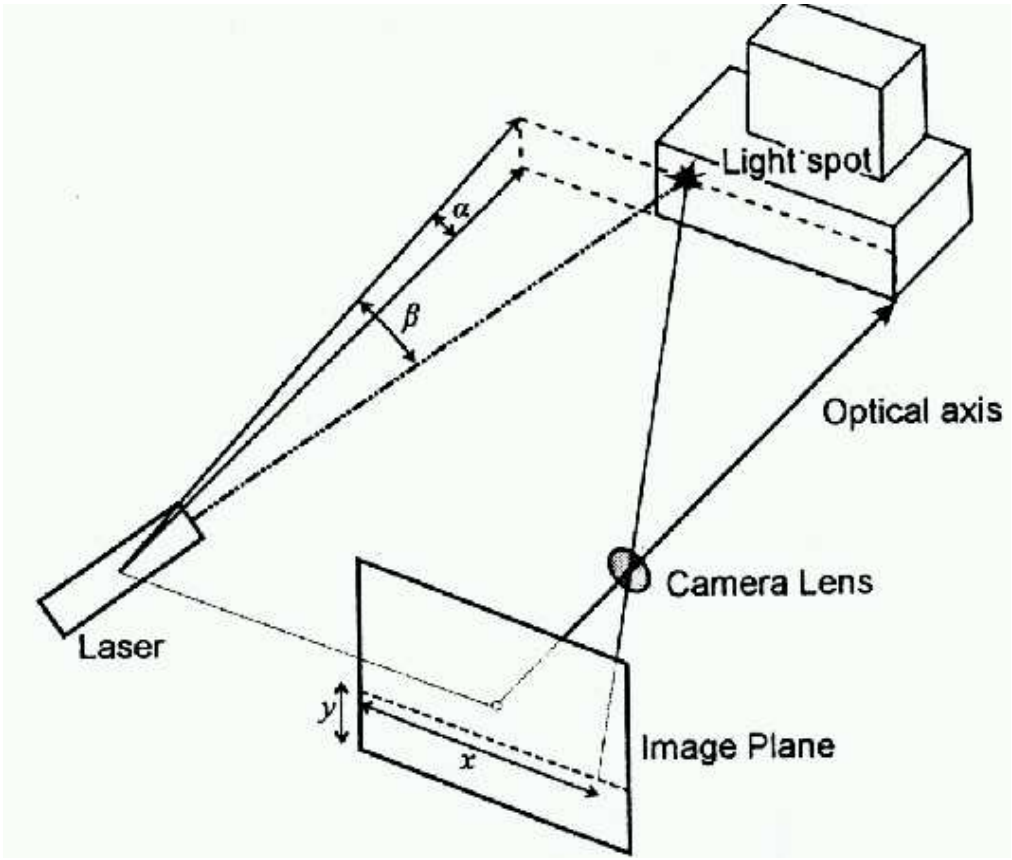
Lumière structurée

Projecteur

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





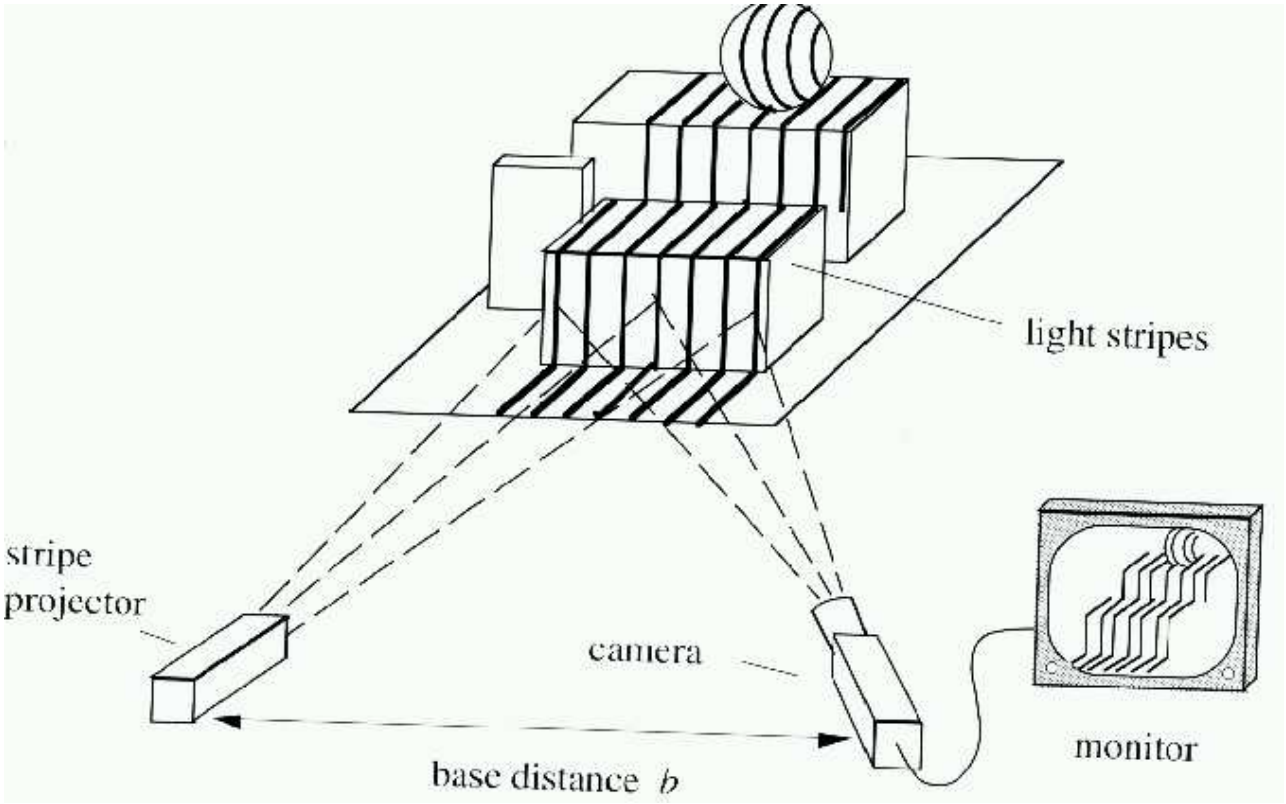


Il suffit de trouver le point laser dans l'image...

Pattern de lumière

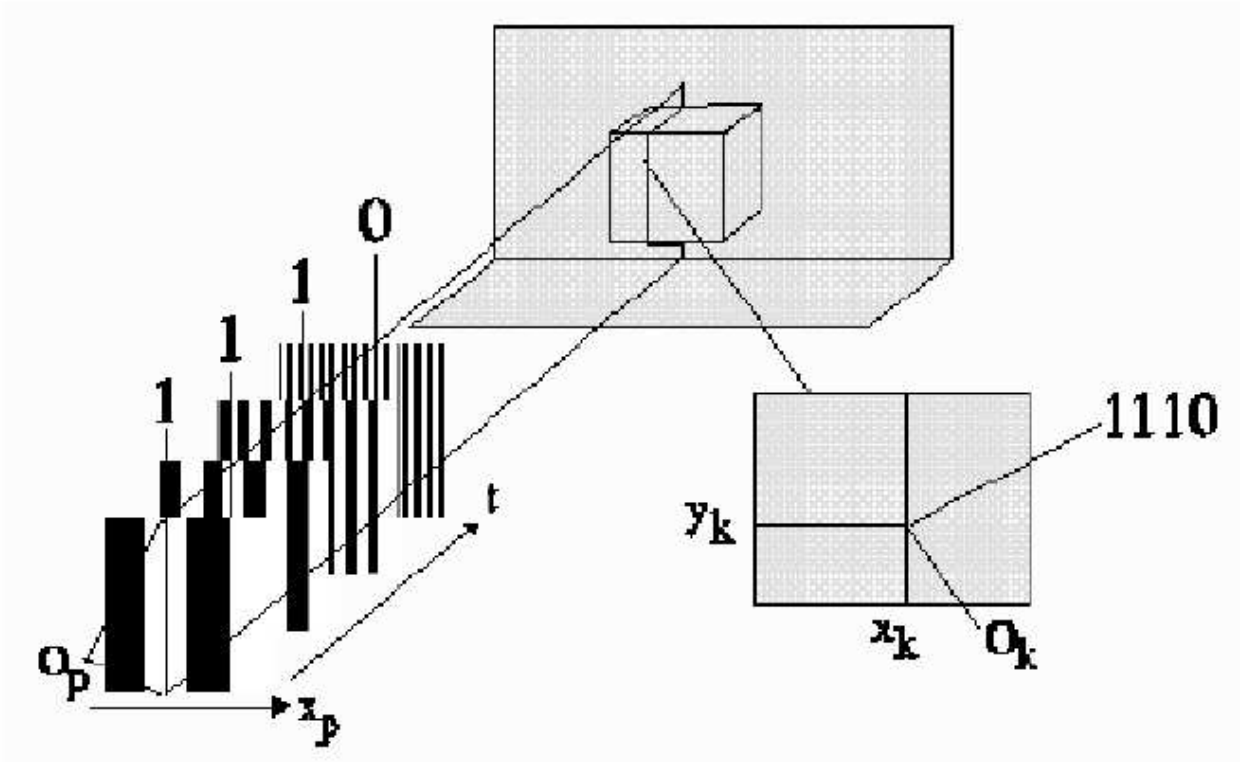
Plusieurs lignes à la fois...

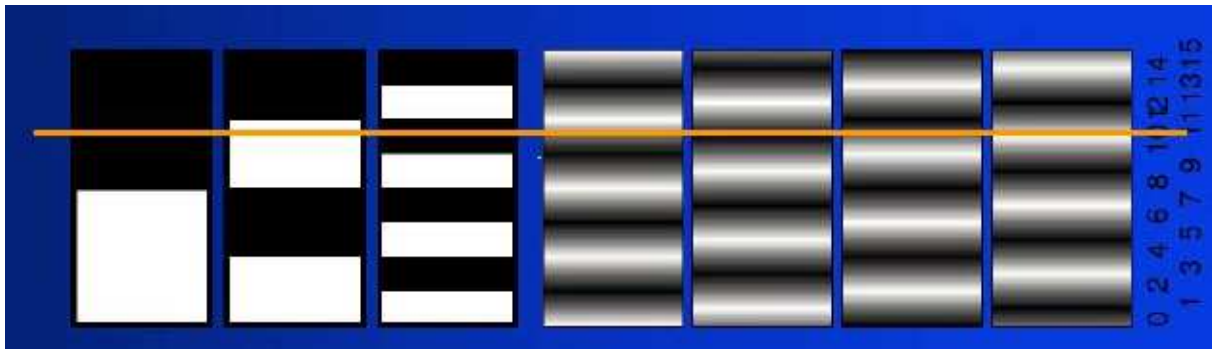
Utiliser un projecteur plutôt qu'un laser !



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.





Comment choisir le code ?

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 $0 \ldots 7$, 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 ?

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

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

- L'inverse s'obtient comme suit :

$$\begin{aligned} N[n-1] &= G[n-1] \\ N[i] &= \mathrm{XOR}(N[i+1], G[i]) \quad i \in [n-2, \ldots, 0] \end{aligned}$$

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.

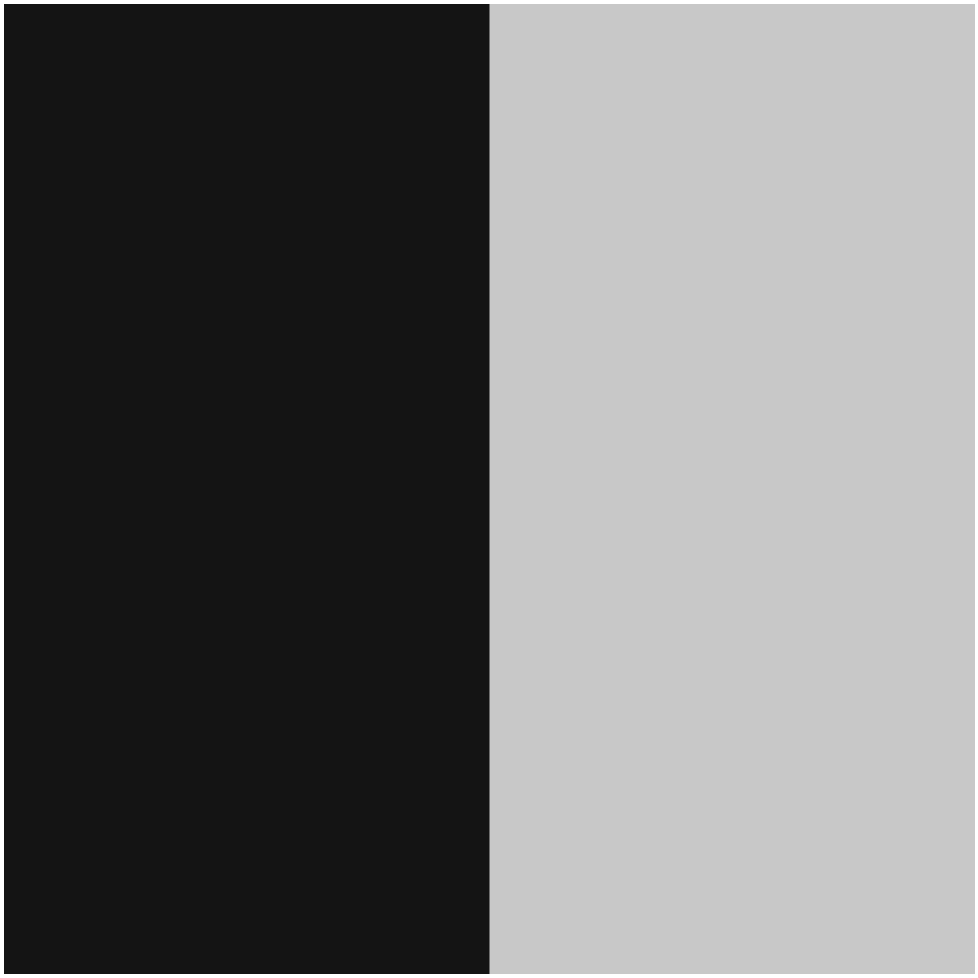
Pour reconstruire

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

Et voilà ! La correspondance est établie.

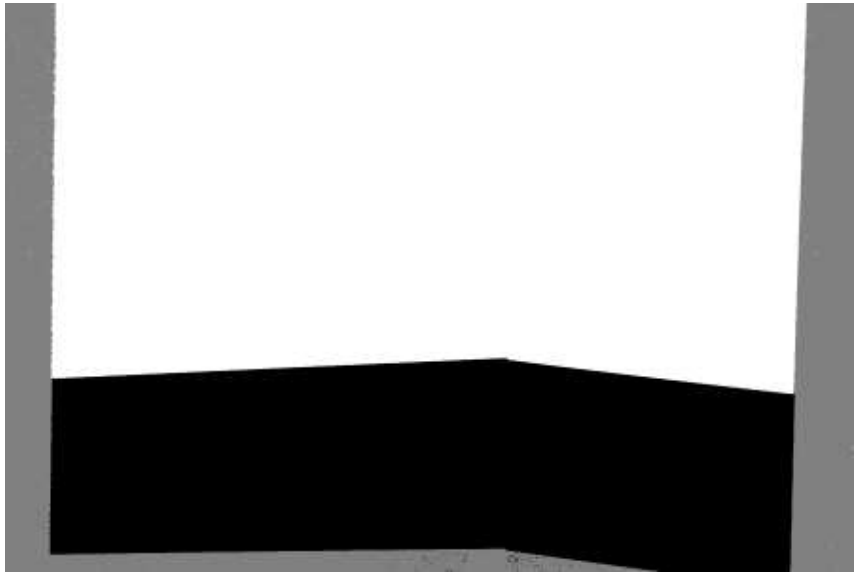
Le point de vue du projecteur

Le projecteur envoie le code...



Le point de vue de la caméra

La caméra regarde le code...



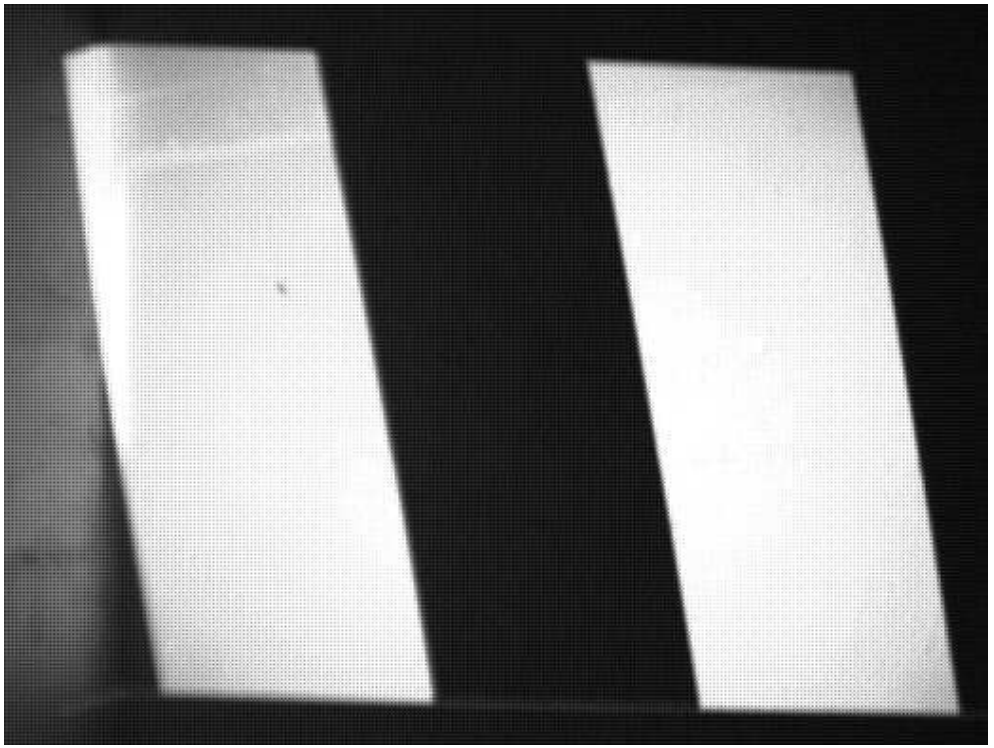
Lumière structurée

On encode les positions en X et Y.

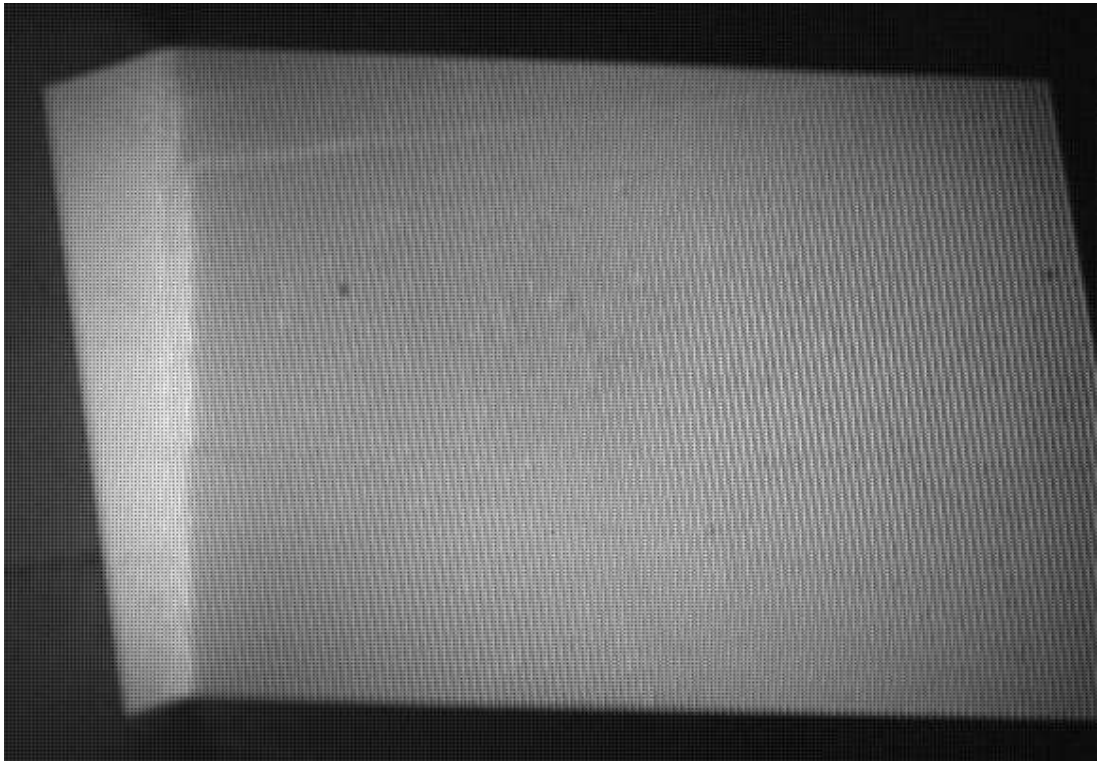
$\Rightarrow$ Positions inconnues de la caméra et du projecteur

Précision des code

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

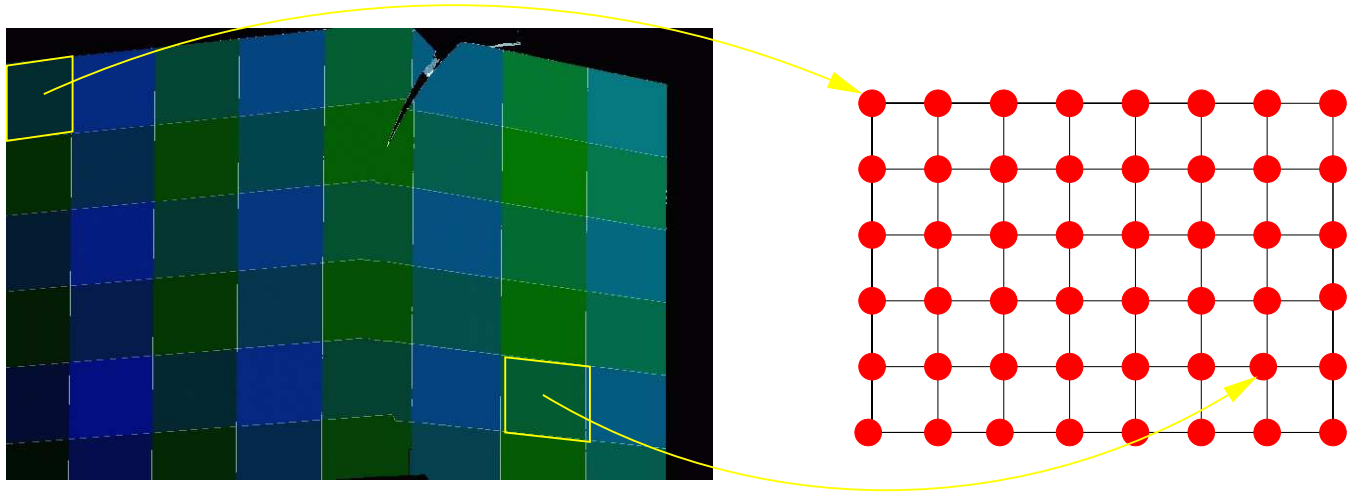


Bits de poids faible



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

Problème de résolution



Camera image

Projector correspondance

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

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

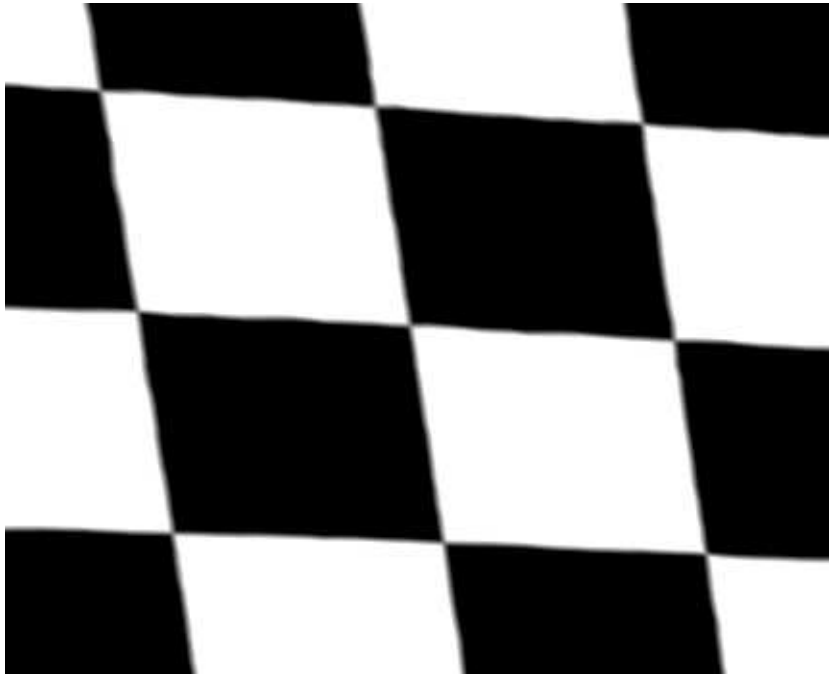
0 1 1 0 0 X X
0 1 1 0 X 1 X
0 1 1 0 X 1 1

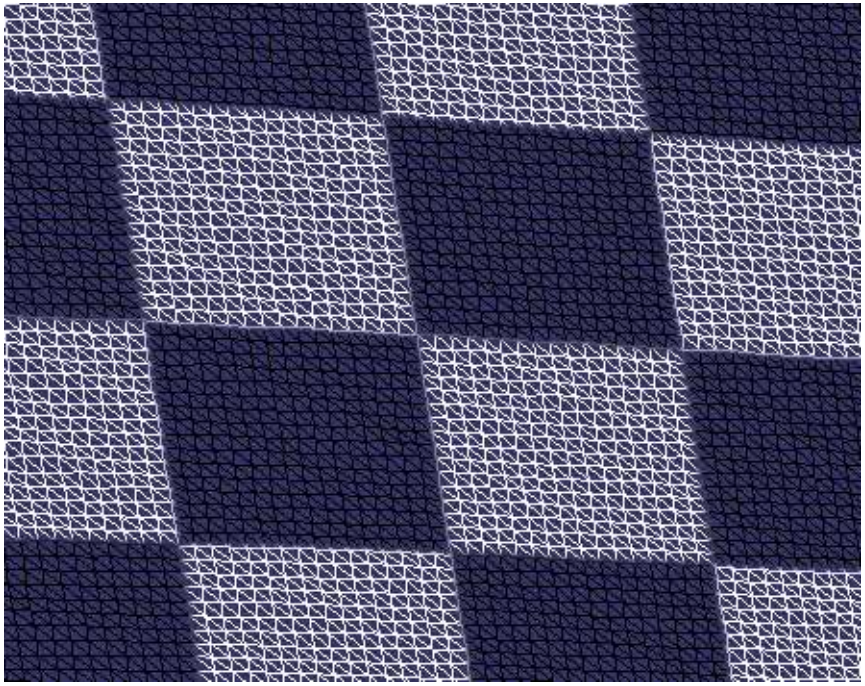
On peut retrouver les bits qui manquent....

Projection distortionnée

Comment projeter...

OpenGL peut afficher une grille et une texture...

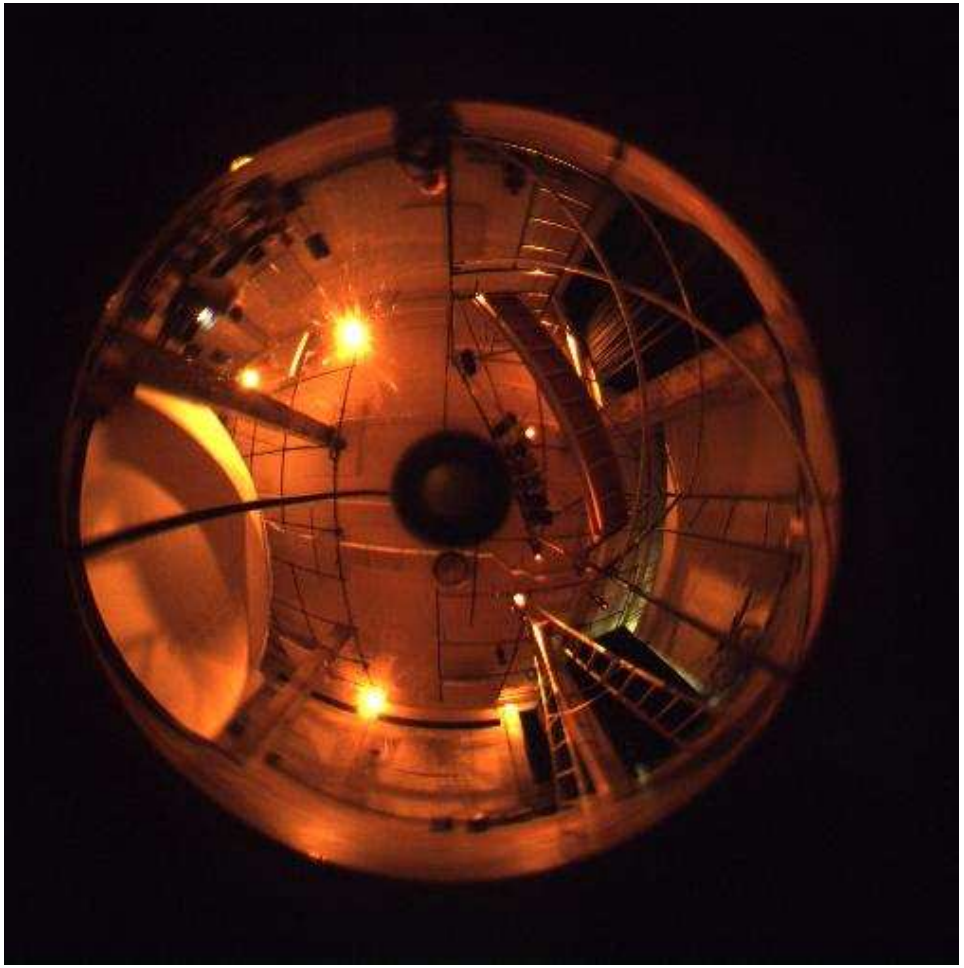




Élimination de la distorsion

Pas de calibration ?

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



Projecteur multiples

Chaque projecteur donne une reconstruction.

Toutes les reconstructions donnent une grille OpenGL.

On peut projeter simultanément à travers tous les projecteurs.

Le futur

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