

Vision par ordinateur: Introduction

Sébastien Roy
Jean-Philippe Tardif

Département d'Informatique et de recherche opérationnelle
Université de Montréal



Hiver 2007

- 1 Organisation du cours
- 2 Introduction à la Vision 3D
- 3 Outils utilisés en vision
- 4 Résumé des cours

Pour la vision 3D :

- Avoir une vue d'ensemble des problématiques
- Découvrir les possibilités offertes par la discipline
- Connaître la théorie à la base du domaine
- Utiliser les outils courants dans l'industrie et la recherche
- Implémenter des algorithmes, les tester et les évaluer

Le cours est une **Introduction** seulement. Nous irons dans les détails pour la base, mais pas pour le reste. Nous essaierons de vous faire connaître l'état de l'art de la discipline pour que vous connaissiez les grands objectifs de la recherche et les possibilités au niveau industriel et artistique.

Professeur Sébastien Roy

Chargé de cours Jean-Philippe Tardif

Bureau 2391 (2384/2262 pour le Lab Vision 3D)

Courriels {roys,tardifj}@iro.umontreal.ca

Téléphone 514-343-6852 ou 514-343-6111 poste 4104

Web (cours) www.iro.umontreal.ca/~roys/fr_ift6145.shtml

Web www.iro.umontreal.ca/~roys

www.iro.umontreal.ca/~tardifj

Horaire Mardi 16h30-18h30,

Jeudi 8h30-10h30, Local Z-200, Pav. C. McNicoll

Laboratoire 2333 (Travaux pratiques)

Un examen intra (20), un examen final (20).

L'évaluation se fait aussi par les travaux pratiques.

%		Durée (semaines)
0	TP 0 : Intro. aux outils de la Vision 3D	1.5
10	TP 1 : Homographie et Mosaïque	2
15	TP 2 : Calibration de caméra	2
15	TP 3 : Stéréo	2
10	TP 4 : Flux optique	2
10	TP 5 : Lumière structurée	2
20	Examen Intra	
20	Examen Final	

Pour chaque travail pratique, il faut :

- remettre un cours rapport écrit
- remettre électroniquement le rapport, les sources, le notebook Mathematica, figures, images, etc...
- être prêt à faire une démonstration "en personne" et à répondre à des questions sur le travail.

Autres détails :

- Les travaux pratiques se font seul.
- Le projet final se fait seul.
- Vous pouvez discuter des problèmes que vous rencontrez avec vos collègues, mais pas de "cut & paste".

Est-ce que je peux utiliser du code de l'internet ? **Oui.**

Conditions d'utilisation

- Il faut **ABSOLUMENT** donner les références.
- Il faut expliquer **CLAIREMENT** votre contribution et celle du matériel trouvé sur l'internet.
- Il faut que vous apportiez une contribution réelle.
Le simple fait de “comprendre” le matériel ne suffit pas.
(on ne peut pas copier un TP de l'an passé, par exemple)

Livre fortement suggéré

David A. Forsyth and Jean Ponce, "Computer Vision : A Modern Approach", Prentice Hall, 2003.

<http://www.cs.berkeley.edu/~daf/book.html>

Autres livres de vision

- R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision", Cambridge University Press, 2nd edition, 2004.
- Y. Ma, S. Soatto, J. Kosecka, S. S. Sastry, "An Invitation to 3-D Vision", Springer, 2005.
- Trucco, Verri, "Introductory Techniques for 3-D Computer Vision", Prentice Hall, 1998.

Logiciels et Langages de programmation

- Mathematica (TP 0,1,2,4)
- C (TP 3,5)

Autres

- GSL (GNU Scientific Library)
- Mathworks MATLAB

Organisation du cours

Leçons	Description
1	Introduction
2	Formation de l'image et Photographie
3	Géométrie projective
4	Révision d'algèbre linéaire ...estimation de transformations, algorithmes robustes
5	Modèle de caméra ...géométrie épipolaire
6	Calibrage de caméra ...méthode directe, planaire, géométrie épipolaire, rotation pure
6	Stéréoscopie
7	Reconstruction multi-vues ...factorisation, auto-calibrage, ajustement de faisceau
8	Flux optique ...estimation du mouvement
9	Lumière structurée
10	Système multi-projecteurs
11	Autres thèmes

Définition :

“Traitement automatisé par ordinateur des informations visuelles.”

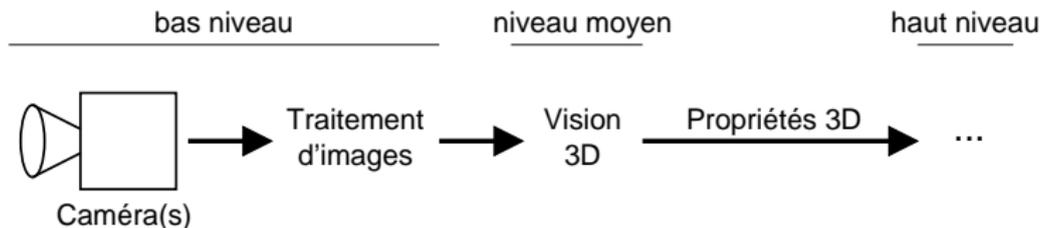
Domaines de recherche :

- Intelligence artificielle
- Robotique
- Traitement de signal
- Psychologie
- Neurologie
- Imagerie médicale

Ce qui nous intéresse :

“Estimer des propriétés 3D du monde à partir d’images.”

- Géométrie (forme, position, orientation, illumination, ...)
- Dynamique (vitesse, trajectoire, ...)
- Reconnaissance (identification, détection, classification, ...)



Traitement d'images (bas niveau)

Première étape pour tout algorithme de vision.

Thèmes :

- Amélioration d'images (dégradation inconnue)
 - Restauration d'images (dégradation connue)
 - Compression (JPEG, MPEG, ...)
 - Extraction de points saillants (coins, contours, ...)
-
- Chapitres 7,8,9 de *Forsyth & Ponce*
 - Chapitres 3,4,5 de *Trucco & Verri*
 - **IFT 6150** Traitement d'images (Max Mignotte)

Reconnaissance (haut niveau)

Détecter, d'identifier, ou de classer les objets, textures, etc...

Thèmes :

- Requier l'utilisation d'information extérieure au système (ex : signalisation routière)
 - Surtout 2D (ex : détecter un biscuit défectueux)
 - Un peu 3D (ex : reconnaissance de visage)
 - 3D trop difficile, requiert l'aide de la Vision 3D
-
- Chapitres 18-24 de *Forsyth & Ponce*
 - Chapitres 10,11 de *Trucco & Verri*
 - **IFT 6141** Reconnaissance de formes (Jean Meunier)

Photogrammétrie

Mesures exactes à partir d'images

Thèmes :

- Ex : Cartes topographique à partir d'images satellites
- Requiert généralement
 - interventions manuelles dans les images
 - interventions manuelles sur le terrain

→ Calibration de caméra, calcul de pose

On classe les problèmes selon le degré d'intervention...

- L'intervention manuelle

Si on peut se permettre d'intervenir manuellement dans les images, la vie devient beaucoup plus facile.

- L'objet de calibration

- Si on peut s'assurer que la scène contient toujours un objet de notre choix, disons un cube, la vie devient beaucoup plus facile.

- Vision 3D

- En général, pas d'objet de calibration ou d'intervention manuelle.

⇒ la vie est beaucoup moins facile.

Infographie

Génération d'images synthétiques à partir de modèles.

Thèmes :

- le plus grand réalisme possible
 - modélisation à partir d'images
-
- **IFT 3350** Infographie (Pierre Poulin)
 - **IFT 6042** Synthèse d'images (Pierre Poulin)
 - **IFT 6095** Art et Science de l'Image (Victor Ostromoukhov)

Optimisation numérique (outil)

- Formulation du problème
- Acquisition des données
- Estimation de modèle \equiv Minimisation de fonction

Thèmes :

- Optimisation de fonctions convexes ou non
- Méthodes linéaires ou non-linéaires
- Problème de graphes
- Méthodes robustes
- Solution continue ou discrète

- **IFT 1063** Mathématique discrètes
- **IFT 2505**
- **IFT 3512** Techniques d'optimisation (1,2)
- **IFT 6542** Flots dans les réseaux
- **IFT 6551** Programmation en nombres entiers
- ...

Quelques exemples :

Inspection et contrôle de qualité
Flot de véhicules routiers et piétons
Vitesse et plaques d'immatriculation
Reconnaissance des expressions faciales
Identification par la pupille
Reconnaissance des gestes
Imagerie médicale 3D (CT,IRM,**fMRI**)
Opérations assistées en neurologie et en orthopédie
Média (recherche dans des bases d'images)

Véhicules autonomes
Robotique intelligente
Applications militaires
Reconnaissance des visages
Réalité virtuelle
Réalité augmentée
Réalité virtualisée (!)
Sécurité
...

- Calcul numérique et symbolique
- Facilite le **prototypage** d'algorithmes numériques
- Langage de programmation fonctionnelle
- Lent (sauf pour les algorithmes numériques déjà fourni, ex : SVD)
- Visualisation
- Exemples :

$$\text{Racines de } 4x^3 - 2x^2 + 3x - 5 \rightarrow \left\{ 1, \frac{-1-i\sqrt{19}}{4}, \frac{-1+i\sqrt{19}}{4} \right\}$$

$$\text{Racines de } ax^2 + bx + c \rightarrow \left\{ \frac{-b+\sqrt{b^2-4ac}}{2a}, \frac{-b-\sqrt{b^2-4ac}}{2a} \right\}$$

Appel récursif : Factoriel...

```
f[0] :=1 ;  
f[i_] :=i*f[i-1] ;
```

D'autres exemples en labo.

- Calcul numérique seulement
- Plus simple que Mathématique, mais moins puissant
- Langage du style Fortran

Simplifie la manipulation des matrices/vecteurs

Exemple :

Code Matlab

$$M1 = [1, 2; 3, 4];$$

$$M1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$M2 \leftarrow [5, 6; 7, 8];$$

$$M2 = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$M12a \leftarrow [M1, M2];$$

$$M12a \rightarrow \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}$$

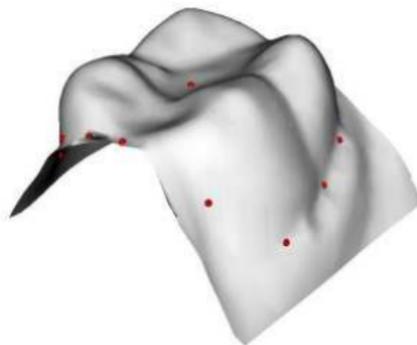
$$M12b = [M1; M2];$$

$$M12b \rightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}$$

VTK : (Visualization Tool Kit)

- Permet de visualiser des données complexes en 3D.
- Interface en Tcl/Tk, C++, Python, Java
- Destiné au prototypage (assez lent)

Par exemple, modèles dentaires :



GNU GSL : (GNU Scientific library)

- Facilite le développement **après prototypage**
- Langage C et C++
- Représentation/manipulation de matrices et vecteurs
- Nombres complexes, polynômes et encore plus
- Algorithmes numériques
- Plus rapide que Mathematica/Matlab
- Mais moins rapide qu'une librairie spécifique à votre problème

Exemple tiré du manuel

(<http://www.gnu.org/software/gsl/manual/>) :

$$\text{Résoudre : } \begin{bmatrix} 0.18 & 0.60 & 0.57 & 0.96 \\ 0.41 & 0.24 & 0.99 & 0.58 \\ 0.14 & 0.30 & 0.97 & 0.66 \\ 0.51 & 0.13 & 0.19 & 0.85 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

...suite

```
#include <stdio.h>
#include <gsl/gsl_linalg.h>

int main (void) {
    double a_data[] = { 0.18, 0.60, 0.57, 0.96,
                       0.41, 0.24, 0.99, 0.58,
                       0.14, 0.30, 0.97, 0.66,
                       0.51, 0.13, 0.19, 0.85 };
    double b_data[] = { 1.0, 2.0, 3.0, 4.0 };

    gsl_matrix_view m = gsl_matrix_view_array (a_data, 4, 4);
    gsl_vector_view b = gsl_vector_view_array (b_data, 4);
    gsl_vector *x      = gsl_vector_alloc (4);
    int s;
    gsl_permutation * p = gsl_permutation_alloc (4);
    gsl_linalg_LU_decomp (&m.matrix, p, &s);
    gsl_linalg_LU_solve (&m.matrix, p, &b.vector, x);

    printf ("x = \n");
    gsl_vector_fprintf (stdout, x, "%g");
    gsl_permutation_free (p);
    return 0;
}
```

Open Source Computer Vision Library

- Facilite le développement **après prototypage**
- Langage C et C++
- Algorithmes de traitement d'images
- Suivi de points
- Calibrage de caméra
- Rapide, mais pas toujours très bien fait
- Bons algorithmes, mais un peu dépassés

<http://www.intel.com/technology/computing/opencv/overview.htm>

Photographie

- Concepts de base pour l'acquisition des images
- Aberrations
- CCD



Espace \mathbb{P}^2 et \mathbb{P}^3

- Représentation et manipulation des éléments : points, lignes, coniques, plans, quadriques.
- Éléments à l'infini
- Transformations
- Projections

Points dans l'espace projectif :

$$\mathbf{p} = [x \quad y]^T, \mathbf{x} \in \mathbb{R}^2$$

$$\mathbf{p} \equiv \mathbf{p}' = \alpha [x \quad y \quad 1]^T, \mathbf{p}' \in \mathbb{P}^2, \forall \alpha \neq 0$$

Outils essentiels

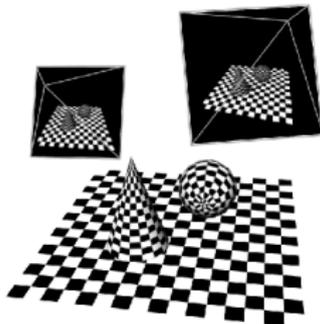
- Norme, produit scalaire/vectorel
- Déterminant, Rang, Inverse d'une matrice
- Décomposition LU, QR, SVD
- Résolution d'un système d'équations linéaires, ou homogène

Exemple :

- Minimiser : $\|A\mathbf{x}\|$
- Solution triviale : $x = \mathbf{0}$ ($\mathbf{0}$ est un vecteur de 0)
- Ajouter la contrainte : $\|x\| = 1$
- Solution par SVD

Objectif

Comment représenter le passage (projection) du monde 3D vers 2D



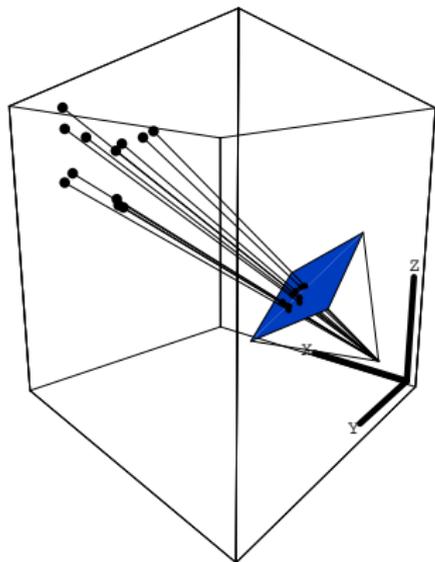
- Choix du modèle (simple vs précis)
- Description des paramètres
- Limitations

Problème :

Estimation des paramètres d'une caméra en utilisant uniquement une ou plusieurs images d'un objet de calibration.

Matrice de projection
classique :

$$\alpha \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = KR[I_{3 \times 3} | -\mathbf{c}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



Objectif

Déterminer la profondeur de chaque pixel d'une image par une mise en correspondance entre deux images

Hypothèses

- Mouvement de caméra connu
- Scène observée est statique
- Changements observés uniquement causés par le mouvement de la caméra



castle-0



castle-6

maximum-flow
(7 images)

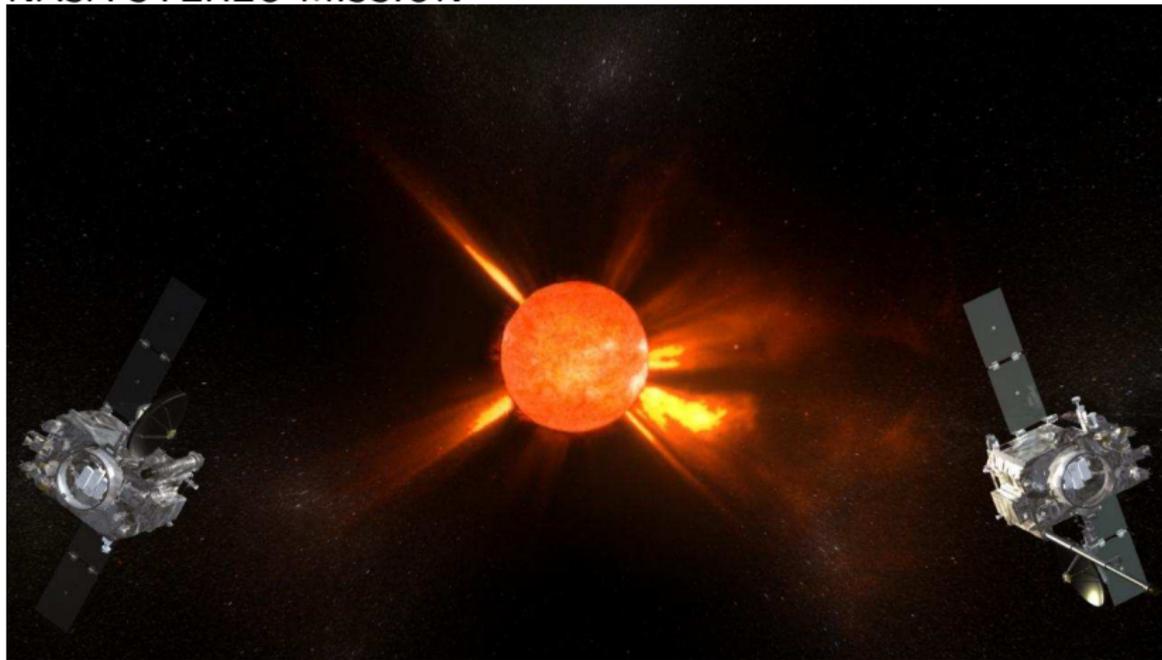


MLMH+V
(13 images)





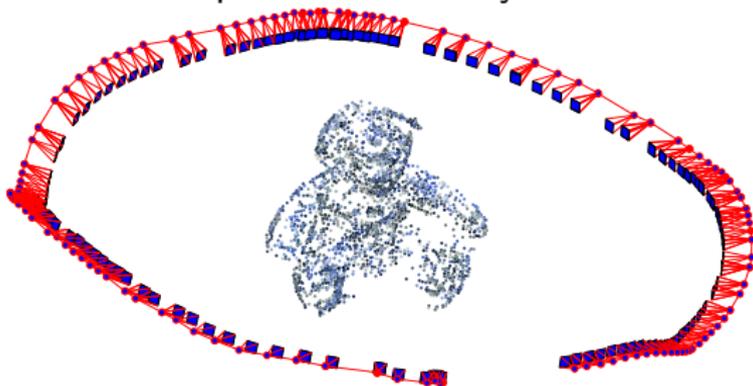
NASA STEREO MISSION



Modèle 3D à partir de suivies de points images

- Factorisation
- Ambiguïté projective ou affine
- Auto-calibrage des caméras
- Ajustement de faisceau (calibré ou non)

Reconstruction de la séquence du "Teddy Bear" :

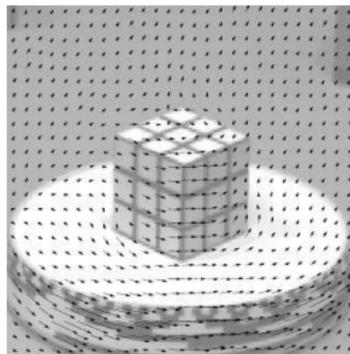
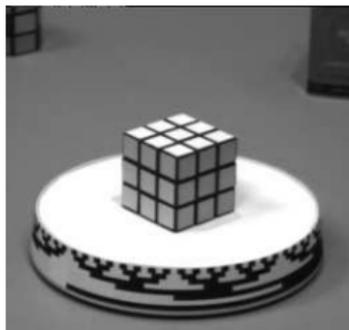


Objectif :

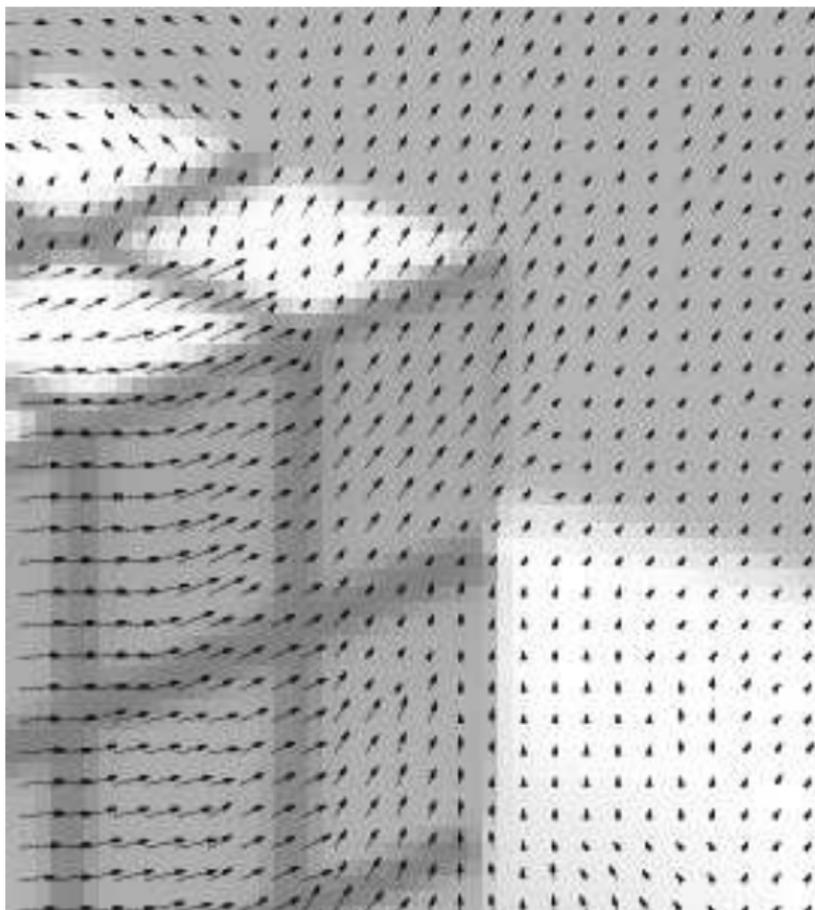
Déterminer le mouvement de chaque pixel d'une image à l'intérieur d'une séquence d'images

Hypothèses

- Mouvement très petit entre chaque image
- Les changements observés sont uniquement causés par le mouvement

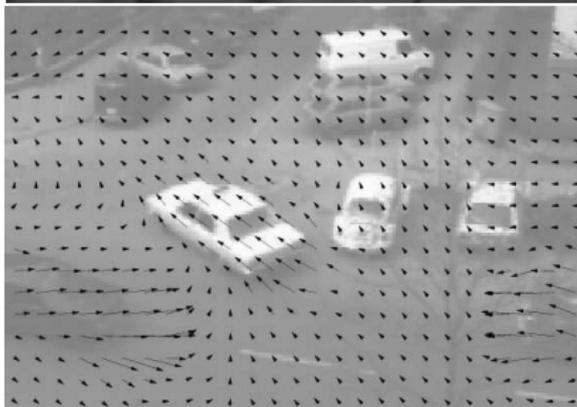


Flux optique



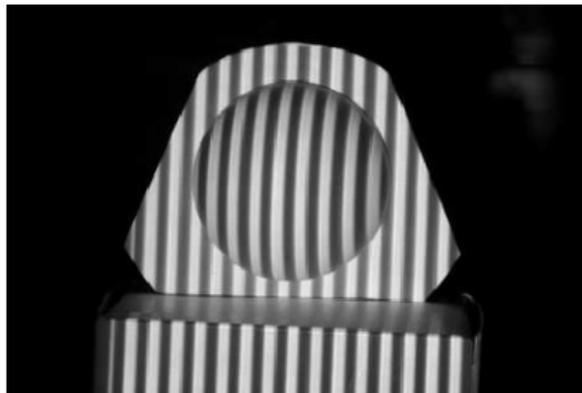
Flux optique

Séquence du Taxi :



Objectif

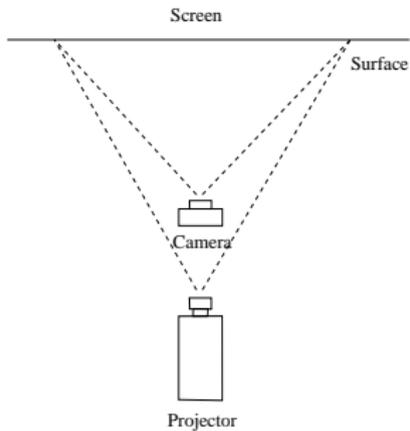
Simplifier la mise en correspondance en remplaçant une caméra par un projecteur de motifs.



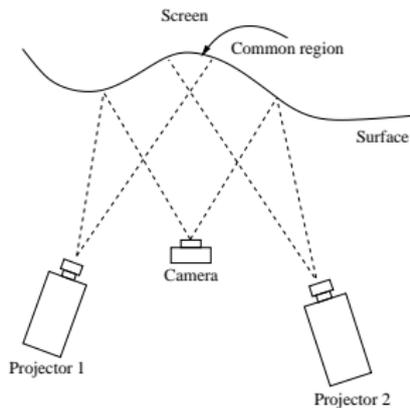
Très utilisé dans les scanners commerciaux

Système multi-projecteurs

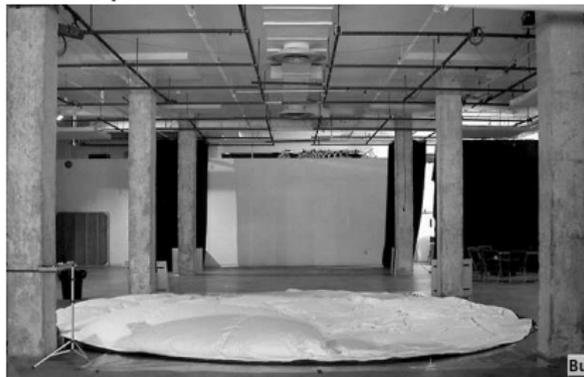
Monde idéal



Réalité



Exemple de surface arbitraire :



Système multi-projecteur

Écran panoramique cylindrique à 9 projecteurs (SAT).



Bon cours !