Reconstruction d'objets convexes à partir de photographies

Présentation de **Lucie-Hélène Cuingnet**

Travail réalisé avec Barnabé Baruchel

TIPE 2025

2025-05-18

Reconstruction 3D

Reconstruction d'objets convexes à partir de photographies

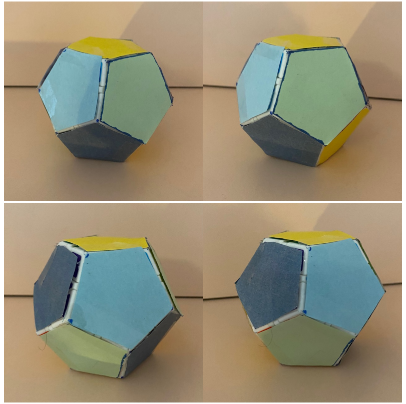
Présentation de Lucie-Hélène Cuingnet

Travail réalisé avec Barnabé Baruchel

TIPE 2025

Pour ce TIPE je suis me suis donnée pour objectif de comprendre et surtout expérimenter comment on peut, à partir de simple photo, retrouver la géométrie complète d'un objet.

Définition du problème

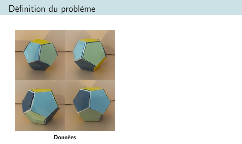


Données

2025-05-18

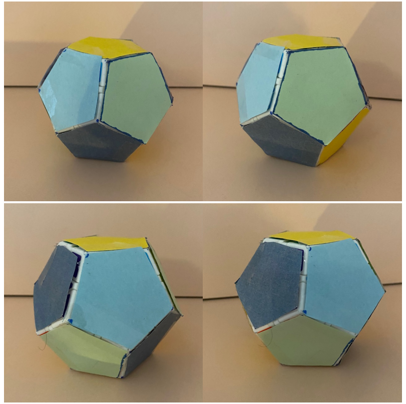
Reconstruction 3D

└Définition du problème

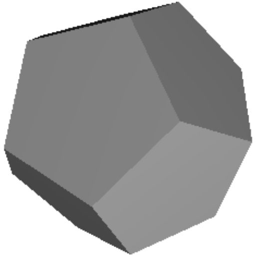


Ainsi, à partir d'un jeux de photo qui constituera mes données je souhaite reconstruire un fichier 3D de ce dernier prêt pour une impression 3D.

Définition du problème



Données

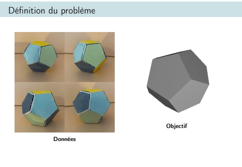


Objectif

2025-05-18

Reconstruction 3D

└Définition du problème



Ainsi, à partir d'un jeux de photo qui constituera mes données je souhaite reconstruire un fichier 3D de ce dernier prêt pour une impression 3D.

Plan

1. Triangulation
2. Analyse des résultats

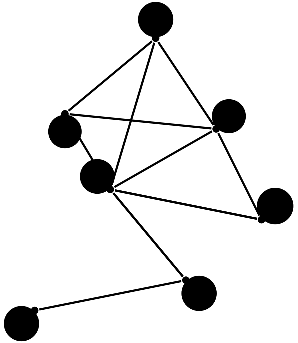
2025-05-18

Reconstruction 3D
└─Triangulation
└─Plan[0.1cm]

Plan

1. Triangulation
2. Analyse des résultats

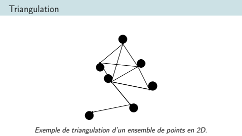
Triangulation



Exemple de triangulation d'un ensemble de points en 2D.

2025-05-18

Reconstruction 3D
└─Triangulation
└─Triangulation



la dernière étape consiste à effectuer une triangulation des points

Plan

1. Triangulation

2. Analyse des résultats
   - Quelques exemples
   - Critiques

Plan
1. Triangulation
2. Analyse des résultats
Quelques exemples
Critiques

Titre d'une slide avant la sous-section

Ici, on n'a pas encore de titre de sous-section dans le bandeau du haut.

Titre d'une slide avant la sous-section

Ici, on n'a pas encore de titre de sous-section dans le bandeau du haut.

Aspect pratique

Ici, on a un titre de sous-section, contrairement à la slide.

Voir le code ici pour référencer une slide avec et la citer avec son numéro via .

Aspect pratique

Ici, on a un titre de sous-section, contrairement à la slide.

Voir le code ici pour référencer une slide avec et la citer avec son numéro via .

Titre sans lettre descendant sous la baseline

Ici c'est mieux, non ?

Titre sans lettre descendant sous la baseline

Ici c'est mieux, non ?

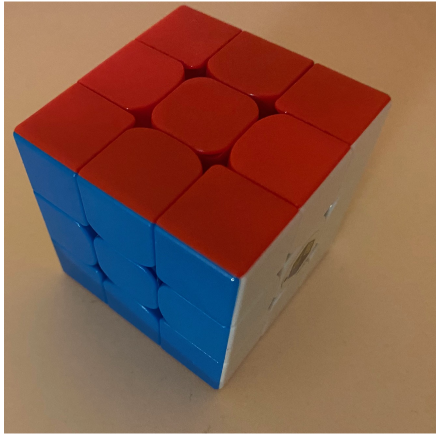
Appendix]Appendix

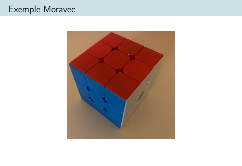
Reconstruction 3D
└─Analyse des résultats
└─Critiques
└─Titre sans lettre descendant sous la baseline

2025-05-18

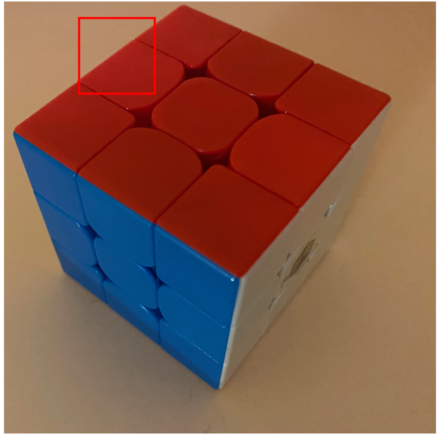
Ici c'est mieux, non ?

Exemple Moravec



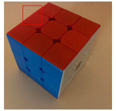


Exemple Moravec



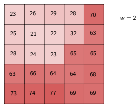
Reconstruction 3D
└─[
2025-05-18
└─Exemple Moravec

Exemple Moravec



23	26	29	28	70
25	21	22	32	63
28	24	23	65	65
63	66	64	64	68
73	74	77	69	69

$w = 2$



Voici une animation de l'évolution de la détection avec l'algorithme de Moravec.
On visualise étape par étape comment les coins apparaissent au fur et à mesure.

Ce genre de visualisation aide à comprendre la sensibilité de l'algorithme selon l'image et les paramètres.

23	26	29	28	70
25	21	22	32	63
28	24	23	65	65
63	66	64	64	68
73	74	77	69	69

$w = 2$

direction horizontal

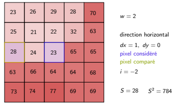
$dx = 1, \; dy = 0$

pixel considéré

pixel comparé

$i = -2$

$S = 28 \qquad S^2 = 784$



Voici une animation de l'évolution de la détection avec l'algorithme de Moravec.
On visualise étape par étape comment les coins apparaissent au fur et à mesure.
Ce genre de visualisation aide à comprendre la sensibilité de l'algorithme selon l'image et les paramètres.

23	26	29	28	70
25	21	22	32	63
28	24	23	65	65
63	66	64	64	68
73	74	77	69	69

$w = 2$

direction horizontal

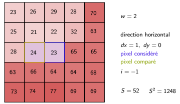
$dx = 1, \ dy = 0$

pixel considéré

pixel comparé

$i = -1$

$S = 52 \qquad S^2 = 1248$



Voici une animation de l'évolution de la détection avec l'algorithme de Moravec.
On visualise étape par étape comment les coins apparaissent au fur et à mesure.
Ce genre de visualisation aide à comprendre la sensibilité de l'algorithme selon l'image et les paramètres.

23	26	29	28	70
25	21	22	32	63
28	24	23	65	65
63	66	64	64	68
73	74	77	69	69

$w = 2$

direction horizontal

$dx = 1, \;\; dy = 0$

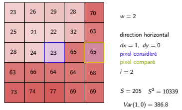
pixel considéré

pixel comparé

$i = 2$

$S = 205 \quad S^2 = 10339$

$Var(1, 0) = 386.8$



Voici une animation de l'évolution de la détection avec l'algorithme de Moravec.
On visualise étape par étape comment les coins apparaissent au fur et à mesure.
Ce genre de visualisation aide à comprendre la sensibilité de l'algorithme selon l'image et les paramètres.

23	26	29	28	70
25	21	22	32	63
28	24	23	65	65
63	66	64	64	68
73	74	77	69	69

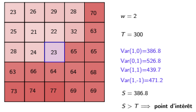
$w = 2$

$T = 300$

Var(1,0)=386.8
Var(0,1)=526.8
Var(1,1)=439.7
Var(1,-1)=471.2

$S = 386.8$

$S > T \implies$ **point d'intérêt**



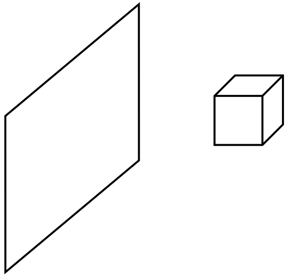
Voici une animation de l'évolution de la détection avec l'algorithme de Moravec.
On visualise étape par étape comment les coins apparaissent au fur et à mesure.
Ce genre de visualisation aide à comprendre la sensibilité de l'algorithme selon l'image et les paramètres.

Voir le schéma détaillé

Voir le schéma détaillé

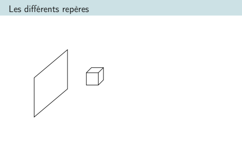
2025-05-18
Reconstruction 3D
└─[

Les différents repères



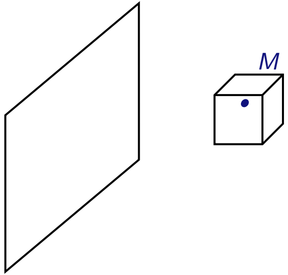
2025-05-18

Reconstruction 3D
└─[
└─Les différents repères



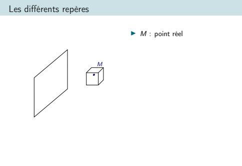
Les différents repères

- M : point réel

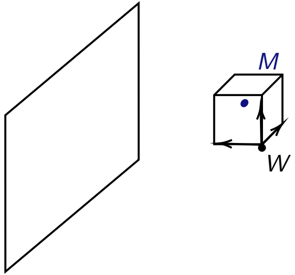


2025-05-18

Reconstruction 3D
└[
└Les différents repères



Les différents repères



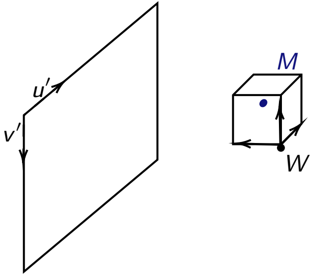
- M : point réel
- W : origine du repère du monde

2025-05-18

Reconstruction 3D
└[
└Les différents repères

Les différents repères
- M : point réel
- W : origine du repère du monde

Les différents repères

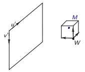


- M : point réel
- W : origine du repère du monde
- (u', v') : coordonnées dans le plan image en pixels

Reconstruction 3D
└[
└Les différents repères

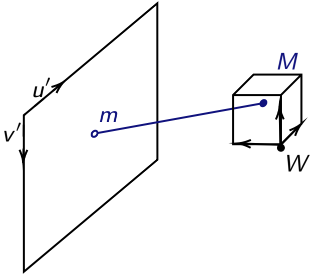
2025-05-18

Les différents repères



- M : point réel
- W : origine du repère du monde
- (u', v') : coordonnées dans le plan image en pixels

Les différents repères

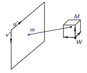


- M : point réel
- W : origine du repère du monde
- (u', v') : coordonnées dans le plan image en pixels
- m : projection de M dans le plan image

2025-05-18

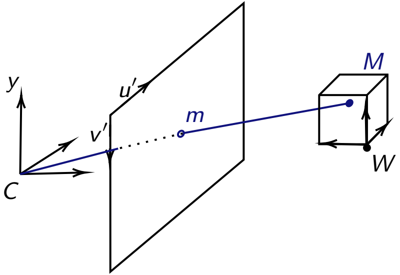
Reconstruction 3D
└[
└Les différents repères

Les différents repères



- M : point réel
- W : origine du repère du monde
- (u', v') : coordonnées dans le plan image en pixels
- m : projection de M dans le plan image

Les différents repères

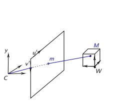


- M : point réel
- W : origine du repère du monde
- (u', v') : coordonnées dans le plan image en pixels
- m : projection de M dans le plan image
- C : origine du repère de la caméra

2025-05-18

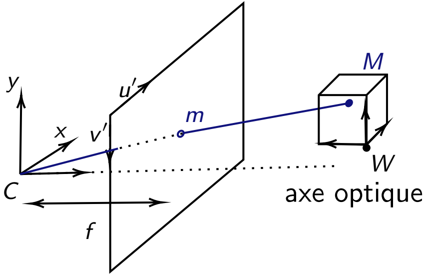
Reconstruction 3D
└[
└Les différents repères

Les différents repères



- M : point réel
- W : origine du repère du monde
- (u', v') : coordonnées dans le plan image en pixels
- m : projection de M dans le plan image
- C : origine du repère de la caméra

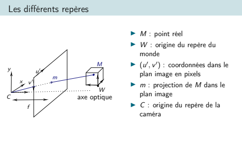
Les différents repères



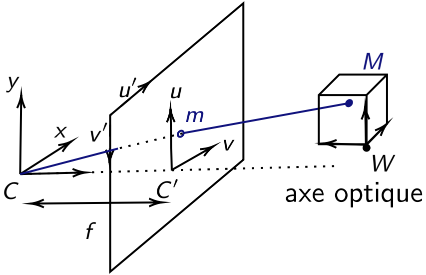
- M : point réel
- W : origine du repère du monde
- (u', v') : coordonnées dans le plan image en pixels
- m : projection de M dans le plan image
- C : origine du repère de la caméra

2025-05-18

Reconstruction 3D
└─[
└─Les différents repères



Les différents repères



- M : point réel
- W : origine du repère du monde
- (u', v') : coordonnées dans le plan image en pixels
- m : projection de M dans le plan image
- C : origine du repère de la caméra
- C' : origine du repère de l'image par projection de C

2025-05-18

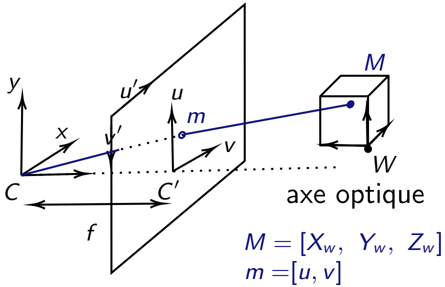
Reconstruction 3D
└[
└Les différents repères

Les différents repères
- M : point réel
- W : origine du repère du monde
- (u', v') : coordonnées dans le plan image en pixels
- m : projection de M dans le plan image
- C : origine du repère de la caméra
- C' : origine du repère de l'image par projection de C

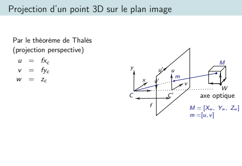
Projection d'un point 3D sur le plan image

Par le théorème de Thalès
(projection perspective)

$$
\begin{aligned}
u &= fx_c \\
v &= fy_c \\
w &= z_c
\end{aligned}
$$



2025-05-18
Reconstruction 3D
└[
└Projection d'un point 3D sur le plan image

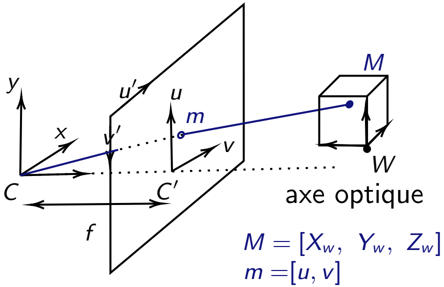


Projection d'un point 3D sur le plan image

Par le théorème de Thalès
(projection perspective)

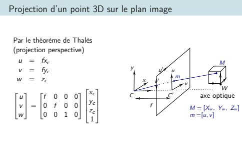
$$\begin{aligned} u &= f x_c \\ v &= f y_c \\ w &= z_c \end{aligned}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$



2025-05-18

Reconstruction 3D
└─[
└─Projection d'un point 3D sur le plan image

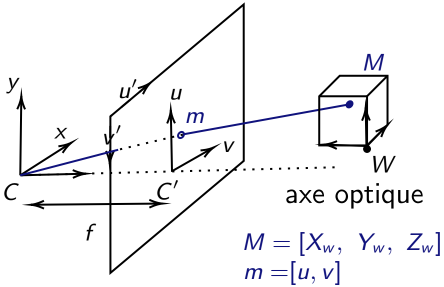


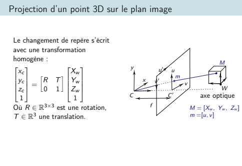
Projection d'un point 3D sur le plan image

Le changement de repère s'écrit avec une transformation homogène :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Où $R \in \mathbb{R}^{3\times3}$ est une rotation, $T \in \mathbb{R}^3$ une translation.





Projection d'un point 3D sur le plan image

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\text{chaîne de projection}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Projection d'un point 3D sur le plan image

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\text{chaîne de projection}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Projection d’un point 3D sur le plan image

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\text{chaîne de projection}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = P \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad \text{avec} \quad P \in \mathcal{M}_{3\times 4}(\mathbb{R})$$

Projection d’un point 3D sur le plan image

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\text{chaîne de projection}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = P \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad \text{avec} \quad P \in \mathcal{M}_{3\times 4}(\mathbb{R})$$

Les différents repères

$$\lambda_i \begin{pmatrix} u^{(i)} \\ v^{(i)} \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} x_C^{(i)} \\ y_C^{(i)} \\ z_C^{(i)} \\ 1 \end{pmatrix}$$

2025-05-18
Reconstruction 3D
└─[
└─Les différents repères

Les différents repères

$$\lambda_i \begin{pmatrix} u^{(i)} \\ v^{(i)} \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} x_C^{(i)} \\ y_C^{(i)} \\ z_C^{(i)} \\ 1 \end{pmatrix}$$

Système d'optimisation à contrainte unitaire

On souhaite résoudre le système en évitant la solution triviale $P = 0$.

Système d'optimisation à contrainte unitaire
On souhaite résoudre le système en évitant la solution triviale $P = 0$.

Système d'optimisation à contrainte unitaire

On souhaite résoudre le système en évitant la solution triviale $P = 0$.
Sachant que la matrice P ne peut être déterminée qu'à un facteur près, on peut imposer :

$$\|P\|^2 = 1$$

et reformuler le système comme un problème d'optimisation :

Système d'optimisation à contrainte unitaire

On souhaite résoudre le système en évitant la solution triviale $P = 0$.
Sachant que la matrice P ne peut être déterminée qu'à un facteur près, on peut imposer :

$$\|P\|^2 = 1$$

et reformuler le système comme un problème d'optimisation :

Système d'optimisation à contrainte unitaire

On souhaite résoudre le système en évitant la solution triviale $P = 0$. Sachant que la matrice P ne peut être déterminée qu'à un facteur près, on peut imposer :

$$\|P\|^2 = 1$$

et reformuler le système comme un problème d'optimisation :

$$\min_{\|p\|^2=1} \|Ap\|^2 = \min_{\|p\|^2=1} p^T A^T A p$$

Système d'optimisation à contrainte unitaire
On souhaite résoudre le système en évitant la solution triviale $P = 0$. Sachant que la matrice P ne peut être déterminée qu'à un facteur près, on peut imposer :
$$\|P\|^2 = 1$$
et reformuler le système comme un problème d'optimisation :
$$\min_{\|p\|^2=1} \|Ap\|^2 = \min_{\|p\|^2=1} p^T A^T A p$$

Système d'optimisation à contrainte unitaire

On souhaite résoudre le système en évitant la solution triviale $P = 0$.
Sachant que la matrice P ne peut être déterminée qu'à un facteur près, on peut imposer :

$$\|P\|^2 = 1$$

et reformuler le système comme un problème d'optimisation :

$$\min_{\|p\|^2=1} \|Ap\|^2 = \min_{\|p\|^2=1} p^T A^T Ap$$

On introduit les fonctions :

- $f(p) = p^T A^T Ap$
- $g(p) = p^T p - 1$

Système d'optimisation à contrainte unitaire

On souhaite résoudre le système en évitant la solution triviale $P = 0$.
Sachant que la matrice P ne peut être déterminée qu'à un facteur près, on peut imposer :

$$\|P\|^2 = 1$$

et reformuler le système comme un problème d'optimisation :

$$\min_{\|p\|^2=1} \|Ap\|^2 = \min_{\|p\|^2=1} p^T A^T Ap$$

On introduit les fonctions :

- $f(p) = p^T A^T Ap$
- $g(p) = p^T p - 1$

Système d'optimisation à contrainte unitaire

On souhaite résoudre le système en évitant la solution triviale $P = 0$. Sachant que la matrice P ne peut être déterminée qu'à un facteur près, on peut imposer :

$$\|P\|^2 = 1$$

et reformuler le système comme un problème d'optimisation :

$$\min_{\|p\|^2=1} \|Ap\|^2 = \min_{\|p\|^2=1} p^T A^T A p$$

On introduit les fonctions :

- $f(p) = p^T A^T A p$
- $g(p) = p^T p - 1$

D'après le théorème d'optimisation sous contrainte (Lagrange), au point optimal P^*, il existe $\lambda \in \mathbb{R}$ tel que :

$$\nabla f(P^*) = \lambda \nabla g(P^*)$$

Système d'optimisation à contrainte unitaire

On souhaite résoudre le système en évitant la solution triviale $P = 0$. Sachant que la matrice P ne peut être déterminée qu'à un facteur près, on peut imposer :

$$\|P\|^2 = 1$$

et reformuler le système comme un problème d'optimisation :

$$\min_{\|p\|^2=1} \|Ap\|^2 = \min_{\|p\|^2=1} p^T A^T A p$$

On introduit les fonctions :

- $f(p) = p^T A^T A p$
- $g(p) = p^T p - 1$

D'après le théorème d'optimisation sous contrainte (Lagrange), au point optimal P^*, il existe $\lambda \in \mathbb{R}$ tel que :

$$\nabla f(P^*) = \lambda \nabla g(P^*)$$

Lien avec les valeurs propres

Posons $M = A^T A$. Alors :

$$f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_i M_{ij} p_j$$

Lien avec les valeurs propres

Posons $M = A^T A$. Alors :

$$f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_i M_{ij} p_j$$

Lien avec les valeurs propres

Posons $M = A^T A$. Alors :

$$f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_i M_{ij} p_j$$

Comme M est symétrique :

$$\frac{\partial f}{\partial p} = 2Mp \quad \text{et} \quad \frac{\partial g}{\partial p} = 2p$$

Lien avec les valeurs propres
Posons $M = A^T A$. Alors :
$$f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_i M_{ij} p_j$$
Comme M est symétrique :
$$\frac{\partial f}{\partial p} = 2Mp \quad \text{et} \quad \frac{\partial g}{\partial p} = 2p$$

Lien avec les valeurs propres

Posons $M = A^T A$. Alors :

$$f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_i M_{ij} p_j$$

Comme M est symétrique :

$$\frac{\partial f}{\partial p} = 2Mp \quad \text{et} \quad \frac{\partial g}{\partial p} = 2p$$

On a donc :

$$\frac{\partial f}{\partial p} = \lambda \frac{\partial g}{\partial p} \quad \Rightarrow \quad \boxed{A^T A p = \lambda p}$$

Lien avec les valeurs propres
Posons $M = A^T A$. Alors :
$$f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_i M_{ij} p_j$$
Comme M est symétrique :
$$\frac{\partial f}{\partial p} = 2Mp \quad \text{et} \quad \frac{\partial g}{\partial p} = 2p$$
On a donc :
$$\frac{\partial f}{\partial p} = \lambda \frac{\partial g}{\partial p} \quad \Rightarrow \quad \boxed{A^T A p = \lambda p}$$

Lien avec les valeurs propres

Posons $M = A^T A$. Alors :

$$f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_i M_{ij} p_j$$

Comme M est symétrique :

$$\frac{\partial f}{\partial p} = 2Mp \quad \text{et} \quad \frac{\partial g}{\partial p} = 2p$$

On a donc :

$$\frac{\partial f}{\partial p} = \lambda \frac{\partial g}{\partial p} \quad \Rightarrow \quad \boxed{A^T A p = \lambda p}$$

C'est une équation aux valeurs propres :

- p est un vecteur propre de $A^T A$
- λ est la valeur propre associée

Lien avec les valeurs propres
Posons $M = A^T A$. Alors :
$$f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_i M_{ij} p_j$$
Comme M est symétrique :
$$\frac{\partial f}{\partial p} = 2Mp \quad \text{et} \quad \frac{\partial g}{\partial p} = 2p$$
On a donc :
$$\frac{\partial f}{\partial p} = \lambda \frac{\partial g}{\partial p} \quad \Rightarrow \quad \boxed{A^T A p = \lambda p}$$
C'est une équation aux valeurs propres :
- p est un vecteur propre de $A^T A$
- λ est la valeur propre associée

Triangulation : formulation du système

- P_1 et P_2 déterminées

2025-05-18
Reconstruction 3D
└─[
└─Triangulation : formulation du système

Triangulation : formulation du système
- P_1 et P_2 déterminées

Triangulation : formulation du système

- P_1 et P_2 déterminées
- On cherche les coordonnées $X = (x_C, y_C, z_C, 1)^T$

Triangulation : formulation du système
- P_1 et P_2 déterminées
- On cherche les coordonnées $X = (x_C, y_C, z_C, 1)^T$

Triangulation : formulation du système

- P_1 et P_2 déterminées
- On cherche les coordonnées $X = (x_C, y_C, z_C, 1)^T$
- Pour chaque paire (x_1, x_2) de projections

2025-05-18

Reconstruction 3D
└─[
└─Triangulation : formulation du système

Triangulation : formulation du système
- P_1 et P_2 déterminées
- On cherche les coordonnées $X = (x_C, y_C, z_C, 1)^T$
- Pour chaque paire (x_1, x_2) de projections

Triangulation : formulation du système

- P_1 et P_2 déterminées
- On cherche les coordonnées $X = (x_C, y_C, z_C, 1)^T$
- Pour chaque paire (x_1, x_2) de projections
- On élimine λ_1, λ_2 et on écrit un système homogène

Triangulation : formulation du système

- P_1 et P_2 déterminées
- On cherche les coordonnées $X = (x_C, y_C, z_C, 1)^T$
- Pour chaque paire (x_1, x_2) de projections
- On élimine λ_1, λ_2 et on écrit un système homogène

Triangulation : formulation du système

- P_1 et P_2 déterminées
- On cherche les coordonnées $X = (x_C, y_C, z_C, 1)^T$
- Pour chaque paire (x_1, x_2) de projections
- On élimine λ_1, λ_2 et on écrit un système homogène
- Système sous la forme $AX = 0$

$$A = \begin{pmatrix} p^1_{31}u_1 - p^1_{11} & p^1_{32}u_1 - p^1_{12} & p^1_{33}u_1 - p^1_{13} & p^1_{34}u_1 - p^1_{14} \\ p^1_{31}v_1 - p^1_{21} & p^1_{32}v_1 - p^1_{22} & p^1_{33}v_1 - p^1_{23} & p^1_{34}v_1 - p^1_{24} \\ p^2_{31}u_2 - p^2_{11} & p^2_{32}u_2 - p^2_{12} & p^2_{33}u_2 - p^2_{13} & p^2_{34}u_2 - p^2_{14} \\ p^2_{31}v_2 - p^2_{21} & p^2_{32}v_2 - p^2_{22} & p^2_{33}v_2 - p^2_{23} & p^2_{34}v_2 - p^2_{24} \end{pmatrix}$$

2025-05-18

Reconstruction 3D

└[

└Triangulation : formulation du système

Triangulation : formulation du système

- P_1 et P_2 déterminées
- On cherche les coordonnées $X = (x_C, y_C, z_C, 1)^T$
- Pour chaque paire (x_1, x_2) de projections
- On élimine λ_1, λ_2 et on écrit un système homogène
- Système sous la forme $AX = 0$

Algorithme: Décomposition QR via Gram-Schmidt

Entrée: $A \in \mathbb{R}^{m \times n}$
Sortie: $Q \in \mathbb{R}^{m \times n}$, $R \in \mathbb{R}^{n \times n}$ tels que $A = QR$

pour $j \leftarrow 1$ **to** n **faire**
- $v_j \leftarrow A_{:,j}$ (*Copie de la $j^{\text{ème}}$ colonne de A*)
- **pour** $i \leftarrow 1$ **to** $j-1$ **faire**
  - $R_{i,j} \leftarrow \langle Q_{:,i}, A_{:,j} \rangle$
  - $v_j \leftarrow v_j - R_{i,j} Q_{:,i}$
- $R_{j,j} \leftarrow \|v_j\|$
- **si** $R_{j,j} > \varepsilon$ **alors**
  - $Q_{:,j} \leftarrow \frac{v_j}{R_{j,j}}$
- **sinon**
  - $Q_{:,j} \leftarrow 0$

retourner Q, R

Algorithme: Décomposition QR via Gram-Schmidt
Entrée: $A \in \mathbb{R}^{m \times n}$
Sortie: $Q \in \mathbb{R}^{m \times n}$, $R \in \mathbb{R}^{n \times n}$ tels que $A = QR$
pour $j \leftarrow 1$ **to** n **faire**
$v_j \leftarrow A_{:,j}$ (*Copie de la $j^{\text{ème}}$ colonne de A*)
pour $i \leftarrow 1$ **to** $j-1$ **faire**
$R_{i,j} \leftarrow \langle Q_{:,i}, A_{:,j} \rangle$
$v_j \leftarrow v_j - R_{i,j} Q_{:,i}$
$R_{j,j} \leftarrow \|v_j\|$
si $R_{j,j} > \varepsilon$ **alors**
$Q_{:,j} \leftarrow \frac{v_j}{R_{j,j}}$
sinon
$Q_{:,j} \leftarrow 0$
retourner Q, R

Algorithme: algorithme QR

Entrée: $B \in \mathbb{R}^{n \times n}$ symétrique
Sortie: Σ^2, V tels que $B = V\Sigma^2 V^T$

$Q_{\text{acc}} \leftarrow I_n$ (*Accumule les produits de Q*)

$\delta \leftarrow 1,\ k_{\max} \leftarrow 1000,\ k \leftarrow 0$
tant que $\delta > 10^{-9}$ et $k < k_{\max}$ **faire**
  $Q, R \leftarrow$ décomposition QR de B
  $B_{\text{nouveau}} \leftarrow R \cdot Q$
  $Q_{\text{acc}} \leftarrow Q_{\text{acc}} \cdot Q$
  $\delta \leftarrow \sum_i |\text{diag}(B_{\text{nouveau}})_i - \text{diag}(B)_i|$
  $A \leftarrow B_{\text{nouveau}}$
  $k \leftarrow k + 1$

pour $i = 1$ à n **faire**
  si $1[i, i] > \varepsilon$ **alors**
    $\Sigma^2[i, i] \leftarrow V[i, i]$
  sinon
    $\Sigma^2[i, i] \leftarrow 0$

retourner Σ^2,Q_{acc}

2025-05-18

Reconstruction 3D
└─[

Algorithme: SVD via algorithme QR sur $A^T A$

Entrée: $A \in \mathbb{R}^{m \times n}$
Sortie: U, Σ, V tels que $A \approx U \Sigma V^T$

$A^T \leftarrow$ transposée de A
$A^T A \leftarrow A^T \cdot A$ (*Symétrique et définie positive*)

algorithme_QR($A^T A$, Σ^2, V) (*Σ^2 diagonale, V orthogonale*)

pour $i \leftarrow 1$ **to** n **faire**
    $\sigma^2 \leftarrow \Sigma^2[i, i]$
    si $\sigma^2 < 10^{-12}$ **alors**
        continuer (*Ignorer valeur singulière nulle*)
    $\sigma \leftarrow \sqrt{\sigma^2}$
    $\Sigma[i, i] \leftarrow \sigma$ (*Met à jour la vraie valeur singulière*)
    $v_i \leftarrow i^{\text{e}}$ colonne de V
    $u_i \leftarrow A \cdot v_i$ (*u_i non normalisé*)
    $u_i \leftarrow u_i / \sigma$
    normaliser u_i
    insérer u_i comme i^{e} colonne de U

2025-05-18

Reconstruction 3D
└[

Algorithme: SVD via algorithme QR sur $A^T A$
Entrée: $A \in \mathbb{R}^{m \times n}$
Sortie: U, Σ, V tels que $A \approx U \Sigma V^T$
$A^T \leftarrow$ transposée de A
$A^T A \leftarrow A^T \cdot A$ (*Symétrique et définie positive*)
algorithme_QR($A^T A$, Σ^2, V) (*Σ^2 diagonale, V orthogonale*)
pour $i \leftarrow 1$ **to** n **faire**
$\sigma^2 \leftarrow \Sigma^2[i, i]$
si $\sigma^2 < 10^{-12}$ **alors**
continuer (*Ignorer valeur singulière nulle*)
$\sigma \leftarrow \sqrt{\sigma^2}$
$\Sigma[i, i] \leftarrow \sigma$ (*Met à jour la vraie valeur singulière*)
$v_i \leftarrow i^{\text{e}}$ colonne de V
$u_i \leftarrow A \cdot v_i$ (*u_i non normalisé*)
$u_i \leftarrow u_i / \sigma$
normaliser u_i
insérer u_i comme i^{e} colonne de U