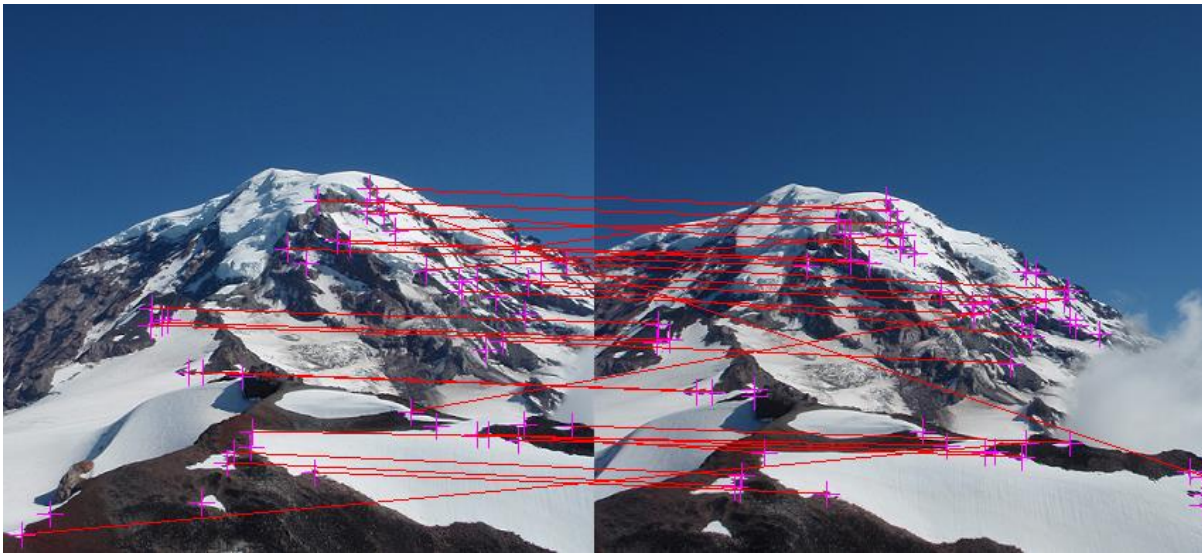


Sujet: Early Jump-Out Corner Detector

Gaspard Langlais

Introduction au sujet

Problématique: Comment détecter les coins dans une image de la manière la plus précise et rapide possible ?



- *Création de panoramas*
- *Détection d'objets en mouvement*

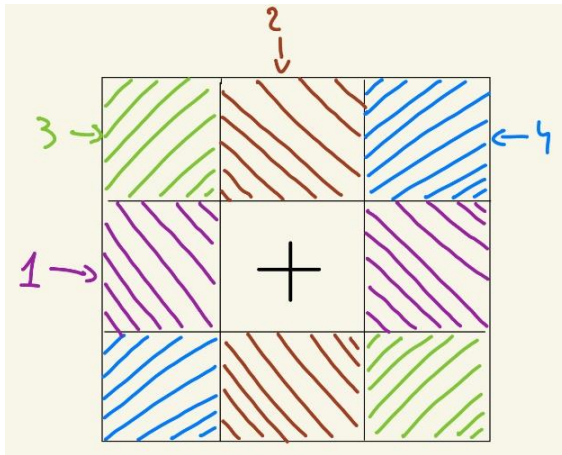
Source : *Ancient Secrets of Computer Vision* by Joseph Redmon

Plan

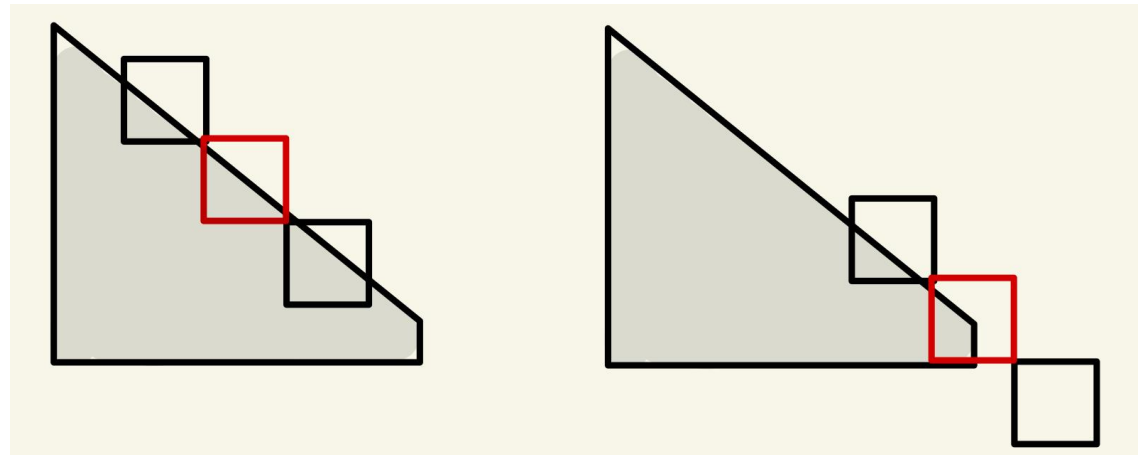
- Technique de détection de coins
- Méthode de Barnea et Silverman
- Amélioration de Cooper
- Présentation des résultats
- Limites de la méthode

Technique de détection de coins

Technique: On utilise une comparaison entre sous fenêtre pour déterminer si un pixel est un coin.



Présentation des 4 directions utilisées



Similaire

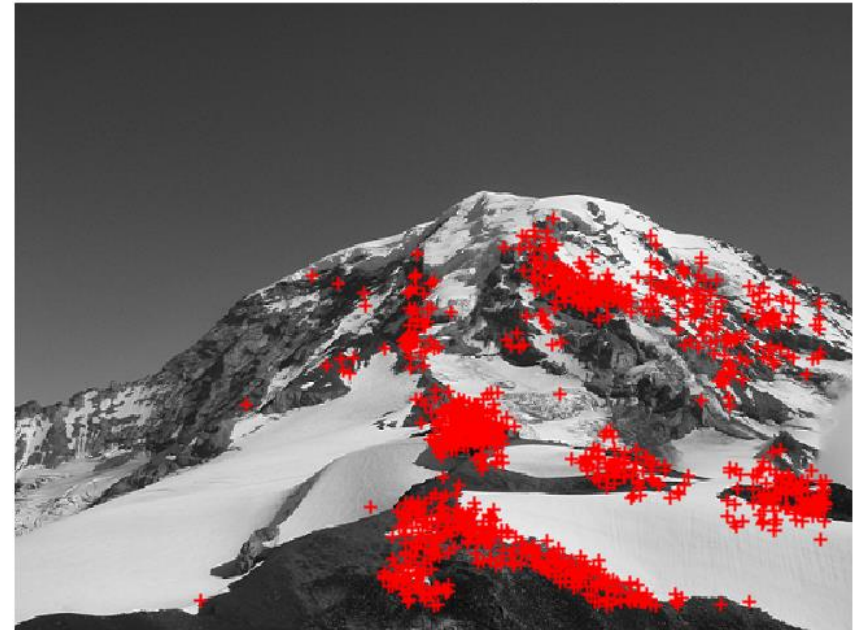
Non Similaire

Méthode de Barnea et Silverman

Méthode: La méthode de Barnea et Silverman optimise les comparaisons de fenêtres en analysant les pixels dans un ordre aléatoire.

Elle utilise un seuil adaptatif qui permet de déterminer si un pixel est un coin sans examiner toute la fenêtre (technique du Jump-Out).

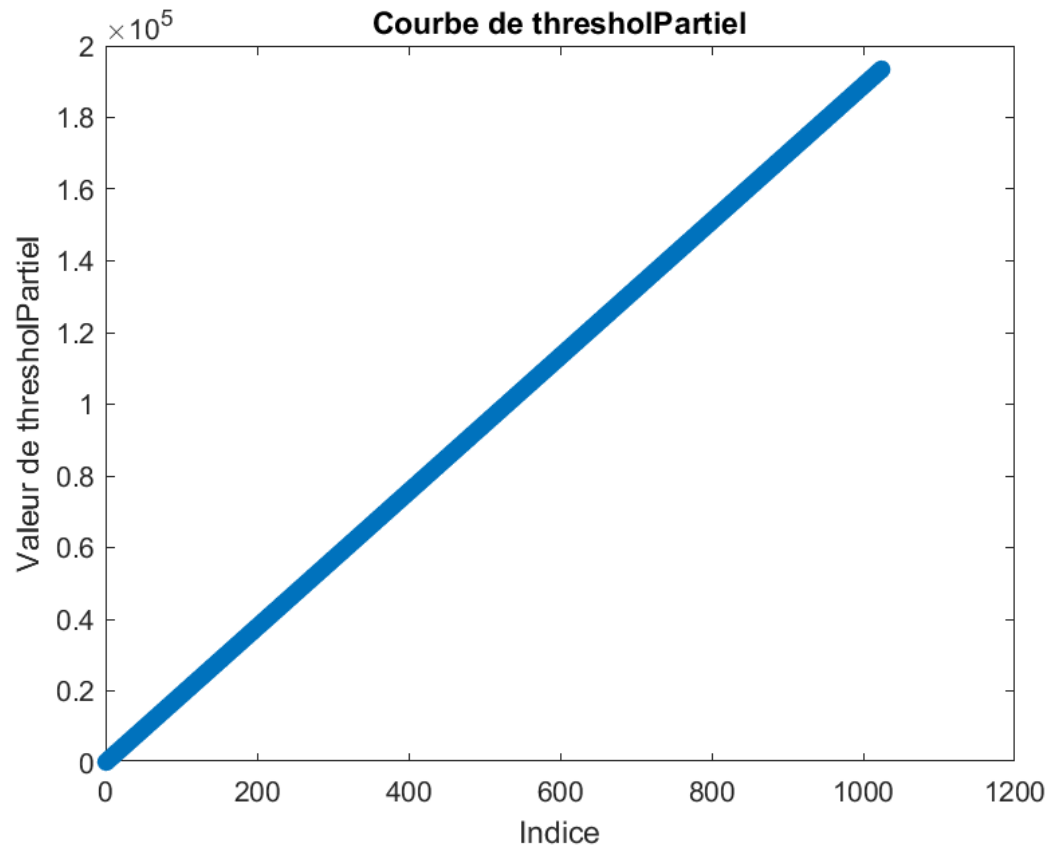
Coins détectés avec Early Jump-Out



Dimensions de l'image: 388 x 517 pixels

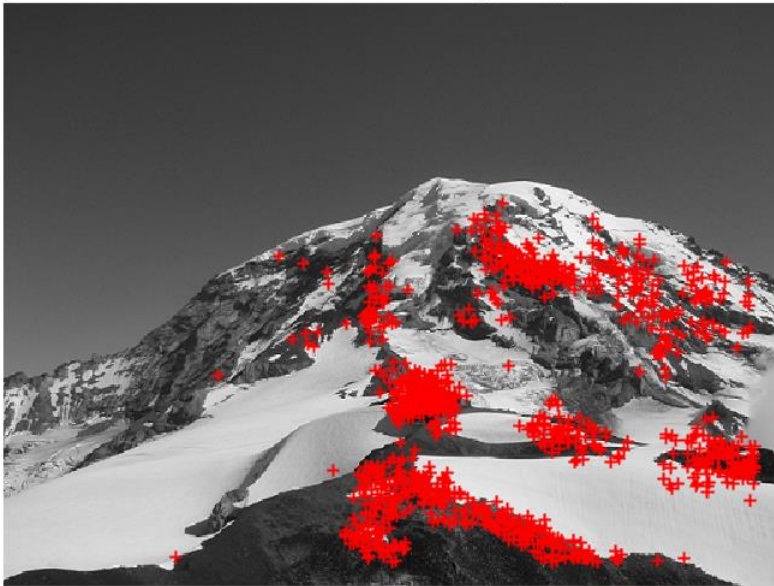
Méthode de Barnea et Silverman: Gestion des seuils

Formule utilisée: $\text{thresholdPartiel} = \text{linspace}(\text{threshold} / \text{windowSize}^2, \text{threshold}, \text{windowSize}^2);$

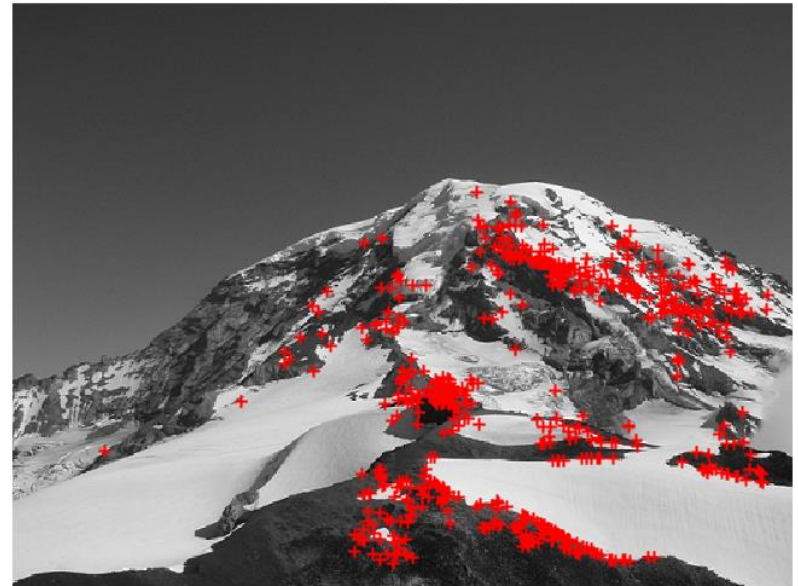


Méthode de Barnea et Silverman: Utilisation des contours

Coins détectés avec Early Jump-Out



Coins détectés avec Early Jump-Out sur les contours



Amélioration supplémentaire: En utilisant les directions contenues dans les contours on peut optimiser le code en déterminant directement la direction dans laquelle la vérification doit être réalisée

Amélioration de Cooper

Méthode: L'amélioration de Cooper permet d'éviter certains faux positifs lors de la détermination de coins.

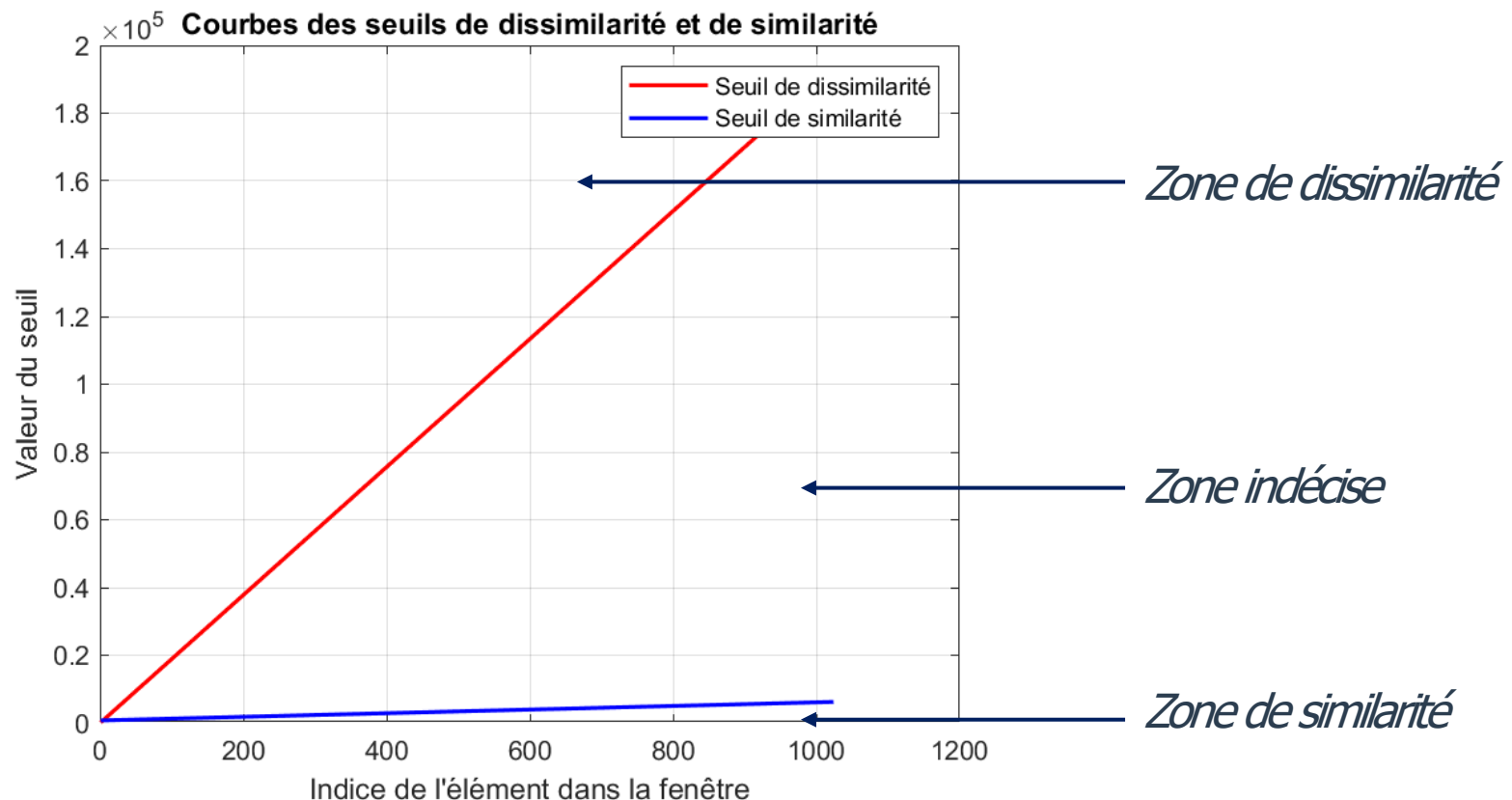
Pour ce faire on ajoute un second un second seuil qui prend en compte la vitesse de croissance de la somme des différences dans les fenêtres.

Coins détectés avec Early Jump-Out et cooper



Amélioration de Cooper: Gestion des seuils

Formule utilisée: $\text{similarityThresholds} = \text{linspace}(\text{threshold} / (2 * \sigma_{\text{noise}}), \text{threshold} / \sigma_{\text{noise}}, \text{windowSize}^2);$



Présentation des résultats

Paramètres utilisés: On utilise une fenêtre de **32*32** et un seuil initial, déterminé expérimentalement de **1700**.

	Détection naïve	Détection avec Barnea-Silverman	Détection avec Barnea-Silverman améliorée	Détection avec Cooper
Fenêtre de 32*32	14 secondes	19.1 secondes	0.6 secondes	0.5 secondes
Fenêtre de 16*16	7.8 secondes	9.5 secondes	0.2 secondes	0.2 secondes

Limites de la méthode

Limites identifiées:

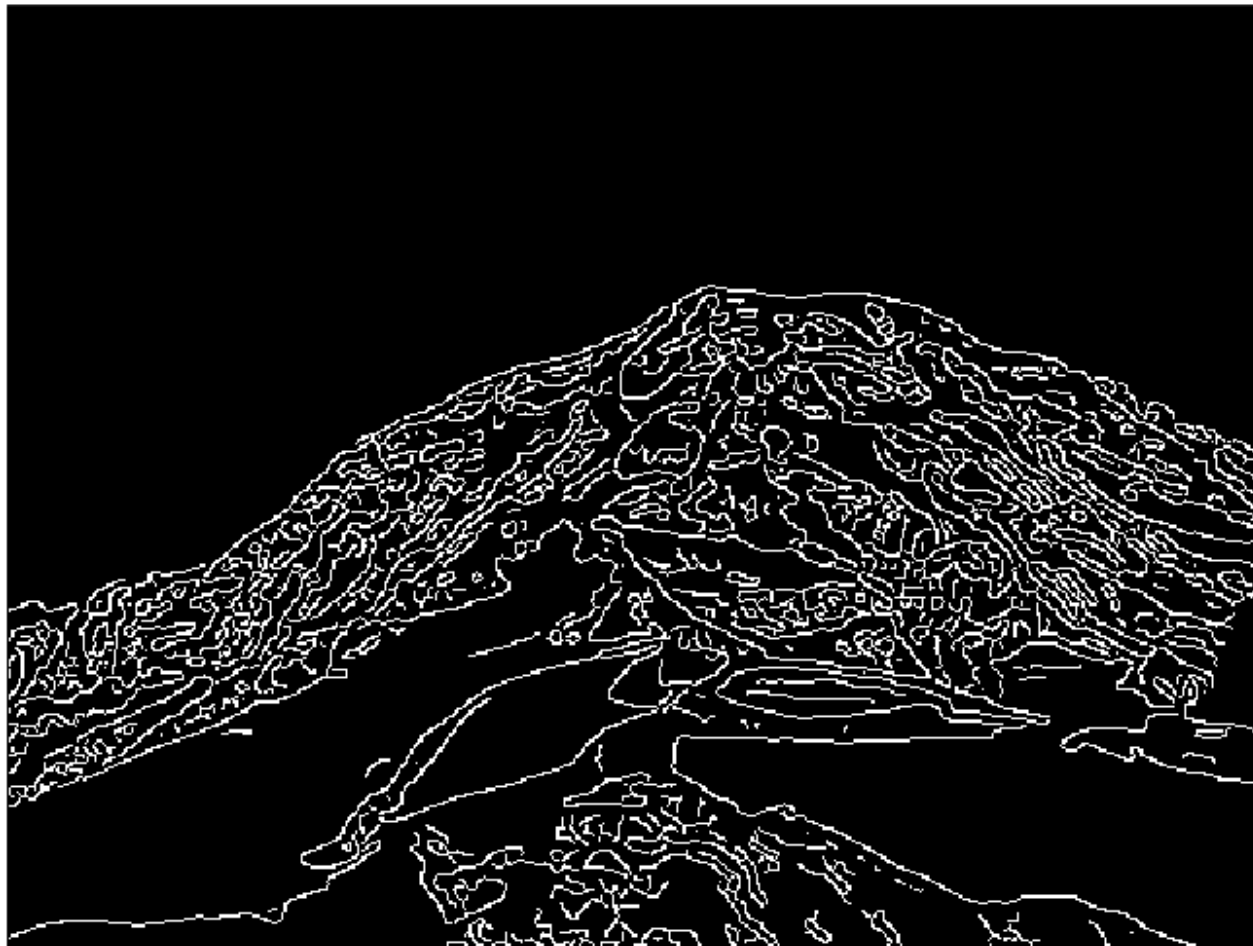
- Nécessité de choisir précisément la taille de fenêtre et le seuil initiale.
- Manque d'optimisation par Matlab de certaines fonctions nécessaires au programme (comme *randperm*).

	Nombre d'appels	Temps utilisé
Détection avec Barnea-Silverman	690 640	14.2 secondes
Détection avec Barnea-Silverman améliorée ou Cooper	15 378	0.4 secondes

*Cas d'une fenêtre 32*32*

Annexe 1: Résultat filtre de Canny

Contours détectés avec l'algorithme Canny



Annexe 2: Méthode de Cooper (16*16)

Coins détectés avec Early Jump-Out et cooper



Annexe 3: Liste des documents utilisés

Liste des sources:

- Source n°1: « Early Jump-Out Corner Detectors »; J.Cooper, S.Venkatesh, L.Kitchen;
https://www.researchgate.net/publication/3192181_Early_jump-out_corner_detectors (document fourni en cours).
- Source n°2: « Early Jump-Out Corner Detectors »; J.Cooper, S.Venkatesh, L.Kitchen;
https://espace.curtin.edu.au/bitstream/handle/20.500.11937/5830/18954_downloaded_stream_46.pdf?sequence=2 (version approfondie).