

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي



جامعة سعيدة. مولاي الطاهر

كلية التكنولوجيا

قسم: الإعلام الآلي

Mémoire de Master

Spécialité : Modélisation Informatique des Connaissances

et du Raisonnement

Thème

Mise en correspondances des
images basées sur les
caractéristiques locales
(étude comparative)

Présenté par :

- RAHALI Linda Abir
- MOHAMEDI Dhiya El
Rahmane

Dirigé par :

Chaibi Hassene



Promotion 2021 - 2022

Table de matières :

Résumé	05
Abstrait	05
ملخص	05
Dédicace	06
Remercîment	07
Introduction général	08
CHAPITRE I : Détection de points d'intérêts	11
1.1. Introduction	12
1.2. Détection d'angle Harris	12
1.3. La détection de points d'intérêts par Hessian	15
1.4. Features from Accelerated Segment Test (FAST)	17
1.5. Détection de points clé à l'aide d'espace d'échelle	20
1.5.1. Harris-Laplace et Hessian-Laplace	20
1.5.2. Détecteur Scale-Invariant Feature Transform (SIFT)	21
1.5.3. Détecteur Speeded-Up Robust Features (SURF)	27
1.5.4. Détecteur orienté FAST et rotatif BRIEF (ORB)	29
1.6. Conclusion	31
CHAPITRE II : Description des caractéristiques locales et la mise en correspondance des points d'intérêts	32
2.1. Introduction	33
2.2. Description des caractéristiques locales par vecteur réel	33
2.2.1. Descripteur Scale-Invariant Feature Transform (SIFT)	34
2.2.2. Descripteur Speeded-Up Robust Features (SURF)	35
2.2.3. PCA-SIFT	35
2.2.4. Gradient Location and Orientation Histogram (GLOH)	37
2.3. Description des caractéristiques locales par vecteur binaire	37
2.3.1. Binary Robust Independent Elementary Features (BRIEF)	37
2.3.2. Binary Robust Invariant Scalable Keypoints (BRISK)	38
2.3.3. Descripteur Oriented FAST and Rotated BRIEF (ORB)	40
2.3.4. Fast Retina Keypoint (FREAK)	41
2.4. Rechercher les points correspondants	42
2.5. Conclusion	43
CHAPITRE III : Expérimentation et résultats	44
3.1. Expérimentations	45
3.1.1. Environnement de développement	45
3.1.2. Les bases d'images utilisées	46
3.1.3. Processus de travail	47
3.2. Résultats et discussion	51
3.2.1. Présentation des images	51
3.2.2. Teste N° 1	51
3.2.3. Teste N° 2	55
3.3. Conclusion	58
Conclusion général et perspectives	59
Référence	60

Liste des figures :

1. Etapes de l'analyse d'image et de la mise en correspondance.	09
1.1. Dérivée première du filtre gaussien.....	13
1.2 Relation entre la distribution des valeurs différentielles dans les coins, les arêtes et les régions plates et les valeurs propres.....	14
1.3. Relation entre les valeurs propres de la matrice des moments du second ordre.....	15
1.4. Filtre gaussien filtre dérivée seconde	16
1.5. Définition des coins dans le détecteur FAST.....	17
1.6. Sélection de pixels sur cercles concentriques par gain d'information.....	19
1.7. Détection des coins par arbre de décision.....	19
1.8. Score de Harris et de Hessian dans l'espace d'échelle.....	21
1.9. Détection extrême à partir d'images DoG	23
1.10. Calcul de l'orientation SIFT.....	26
1.11. Approximation du filtre Gaussien en dérivée seconde par le filtre Box..	27
1.12. Recherche de valeurs extrêmes à l'aide du filtre Box.....	29
1.13. Calcul de l'orientation SURF	29
1.14. Acquisition d'échelle par pyramide d'images.....	30
2.1. Description des caractéristiques SIFT.....	34
2.2. Description des caractéristiques SURF.....	35
2.3. Description des caractéristiques de PCA-SIFT.....	36
2.4. Description des fonctionnalités par GLOH.....	37
2.5. Modèle de paire de pixels pour les caractéristiques binaires.....	38
2.6. Paires BRISK à longue et courte portée.	39
3.1. Un échantillon de tissu de cinq coupes de tache avec des points de repère en surbrillance	47
3.2. La méthodologie de test	48
3.3. Test1 du descripteur BRISK	51
3.4. Test1 du descripteur ORB	52
3.5. Test1 du descripteur SIFT	53
3.6. Test1 du descripteur SURF.....	53
3.7. Resultat de Test 1 utilisant FLANN.....	54
3.8. Resultat de Test 1 avec Brute Force	54
3.9. Points clé pour teste 2	56
3.10. Test2 du descripteur BRISK	56
3.11. Test2 du descripteur ORB.....	56
3.12. Test2 du descripteur SIFT	57
3.13. Test2 du descripteur SURF.....	57
3.14. résultats du Test N°2 ; différence entre BF et Flann	58

Liste des tableaux :

3.1.Description de matériel	45
3.2.Description de logiciel	45
3.3.Distance BF matching pour descripteurs des caracteristiques.....	49
3.4.Comparaison de distance moyenne entre BF et FLANN.....	55
3.5.Comparaison de temps d'exécution entre BF et FLANN.....	55
3.6.résultats du Test N°2 , BF et Flann	57

ملخص

تُستخدم طرق رؤية الكمبيوتر في الوقت الحاضر في العديد من التطبيقات، على سبيل المثال المراقبة بالفيديو أو التصوير الطبي أو إعادة البناء ثلاثي الأبعاد. بحيث تركز هذه الأخيرة على عمليات التعرف على الأشكال و النماذج. للقيام بذلك، يتم تحليل الصورة لاستخراج المعالم أو العناصر الأولية (المحيط أو وظائف شدة الإضاءة أو النماذج المورفولوجية). تعتمد الطرائق الحالية على استعمال النقاط الأساسية التي تمثل عدم استمرارية تدرجات الرمادي التي دورها تحدد الأركان أو الزوايا الموجودة في الصورة. يتم استعمال توصيف محلي من أجل القيام بمطابقة مجموعة من النقاط الأساسية من صورة لصورة أخرى. تسمح هذه التقنية باستخراج معلومات حول المنطقة المجاورة لكل نقطة (قيم البيكسل و شدة الإضاءة و الميل). يهدف هذا العمل البحثي المتمثل في هذه الأطروحة إلى إعطاء نظرة حول مختلف الأدوات المستعملة للكشف عن النقاط الأساسية ووصفها ومطابقتها. أولا نقترح تحليلا ببيولوجرافيا للطرق الموجودة حاليا , بحيث تمكننا هذه الأخيرة من استنتاج نهج التنفيذ بالإضافة إلى المزايا والعيوب الرئيسية. بعد ذلك نتحدث بالتفصيل حول الطريقة المقترحة من أجل تقييم فاعلية الخوارزميات المستعملة. يتم استخدام قاعدتي بيانات تصوير طبي للأنسجة في التجارب المقدمة. الخوارزميات المستعملة للكشف عن العناصر الأولية ووصفها هي (SIFT , SURF, BRISK, ORB, BRIEF , FREAK et FAST). أما بالنسبة لإيجاد التناظر بين النقاط الرئيسية فأساسه الطريقتين BF-Matcher و Flann .

الكلمات المفتاحية : مطابقة, توصيف, BF-Matcher, Flann, علم الانسجة

Abstract

Today computer vision methods are used in many applications such as video monitoring, medical imaging or 3D reconstruction. These different applications generally rely on pattern recognition or tracking processes. To do this, the image is analyzed in order to extract landmarks or primitives (contours, intensity functions or morphological models). The most common methods are based on the use of key points representing a discontinuity of gray levels characterizing a in an image. In order to match a set of points from one image to another, a local description is used. It makes it possible to extract information from the neighborhood of each point (values of pixels, light intensities, gradients).

The research work presented in this thesis aims to implement different tools for detecting, describing and matching key points. First, we propose a bibliographical analysis of existing methods that makes it possible to deduce the implementation parameters as well as the main advantages and disadvantages.

We then detail the process proposed for evaluating the performance of the algorithms used. Two histology medical imaging datasets are used for the tests. The detection and description of the primitives is based on the algorithms (SIFT, SURF, BRISK , ORB, BRIEF , FREAK and FAST). The matching of key points is based on the BF-Matcher and Flann methods.

Keywords : Matching, Descriptors, BF-Matcher, FLANN, Histology

Résumé

De nos jours les méthodes de vision par ordinateur sont utilisées dans de nombreuses applications telles que la vidéosurveillance, l'imagerie médicale ou la reconstruction 3D par exemple. C'est différentes applications s'appuient généralement sur des procédés de reconnaissance de formes ou de suivi. Pour ce faire, l'image est analysée afin d'en extraire des amers ou des primitives (contours, fonction d'intensité ou modèles morphologiques). Les méthodes les plus courantes s'appuient sur l'utilisation de points d'intérêts représentant une discontinuité des niveaux de gris caractérisant un coin dans une image. Afin de mettre en correspondance un ensemble de points d'image à une autre, une description locale est utilisée. Elle permet d'extraire l'information du voisinage de chaque point (valeur des pixels, des intensités lumineuses, des gradients). Les travaux de recherche présentée dans ce mémoire ont pour objectif de mettre en œuvre différents outils de détection, description et mise en correspondance de points d'intérêts. Dans un premier temps, nous proposons une analyse bibliographique de méthodes existantes. Cette dernière permet en effet déduire les paramètres de mise en œuvre ainsi que les principaux avantages, et inconvénients. Nous détaillons par la suite le processus proposé pour évaluer des performances des algorithmes utilisé. Deux bases d'images médicales sont utilisées pour les tests. La détection et le description des primitives repose sur les algorithmes (SIFT, SURF, BRISK, ORB, BRIEF, FREAK et FAST) La mise en correspondance des points d'intérêts se base quant a elle sur les méthodes BF-Matcher et FLANN.

Mots clé : Mise en correspondance, Descripteurs, BF-Matcher, FLANN, Histologie

Dédicace :

Aux élèves de primaires.

Remerciement :

Beaucoup de gens ont nous aidé à réaliser cette thèses, et avoir ce diplôme. D'abord et avant tout, notre directeur de mémoire Hassen Chaibi, nous vous adressons nos plus vifs remerciement pour nous encadrer, orienter, aider et conseiller.

Nos professeurs et enseignants du premier jour à l'école jusqu'au aujourd'hui. Sans vos efforts nous ne soyons pas ici, écrivant ces lignes, MERCI beaucoup.

Nos familles, et amis de leur support et prières.

Nos remerciements vont également aux membres du jury d'avoir accepté d'évaluer notre travail.

Mais Le grand remerciement est dédié à Allah parce que c'est lui qui nous a donnait le savoir et le pouvoir à terminer ce travail. Alhamdulillah.

Introduction Général

Contexte :

De nos jours, l'imagerie numérique devient de plus en plus présente dans les applications courantes de la vie. Elle permet par exemple de surveiller, de localiser, de reconnaître, ou encore de diffuser des informations. Depuis des dizaines d'années les scientifiques cherchent et proposent des procédés afin d'acquérir ou de créer de telles images, de les interpréter, de modifier leur contenu, ou encore d'en extraire l'ensemble des informations nécessaires à diverses applications.

Pour en donner quelques exemples, nous pouvons citer la reconstruction 3D permettant de modéliser, à partir d'un certain nombre d'images, l'environnement entourant la ou les caméras, Il existe également des procédés de suivi d'objets, nécessitant une analyse préliminaire (détection et description) afin de caractériser les points extraits des images. Le suivi (ou *tracking*), est défini par l'étude du déplacement au fil du temps de points d'intérêt, dont l'identification se base sur leur descripteur. D'autres procédés tels que le calibrage de caméras ou la stéréovision sont amenés à utiliser des outils de caractérisation de points et de mise en correspondance. L'utilisation des images médicales, s'appuyant également sur ce type d'outil. Les différentes applications citées précédemment s'appuient sur un ensemble de données extraites de l'image. Généralement la première étape consiste à identifier les régions présentant des propriétés locales remarquables (contours, points, textures). Les applications utilisent principalement des points d'intérêt, définissant une double discontinuité de la fonction d'intensité. Les points extraits sont en second lieu caractérisés, le plus souvent localement, afin d'analyser l'information présente dans le masque de description (forme géométrique définissant une région d'intérêt). Une dernière étape permet d'apparier les points d'intérêt présents dans deux images et d'extraire ainsi des couples. La figure 1 schématise les différentes étapes de l'analyse pour une mise en correspondance de deux images.

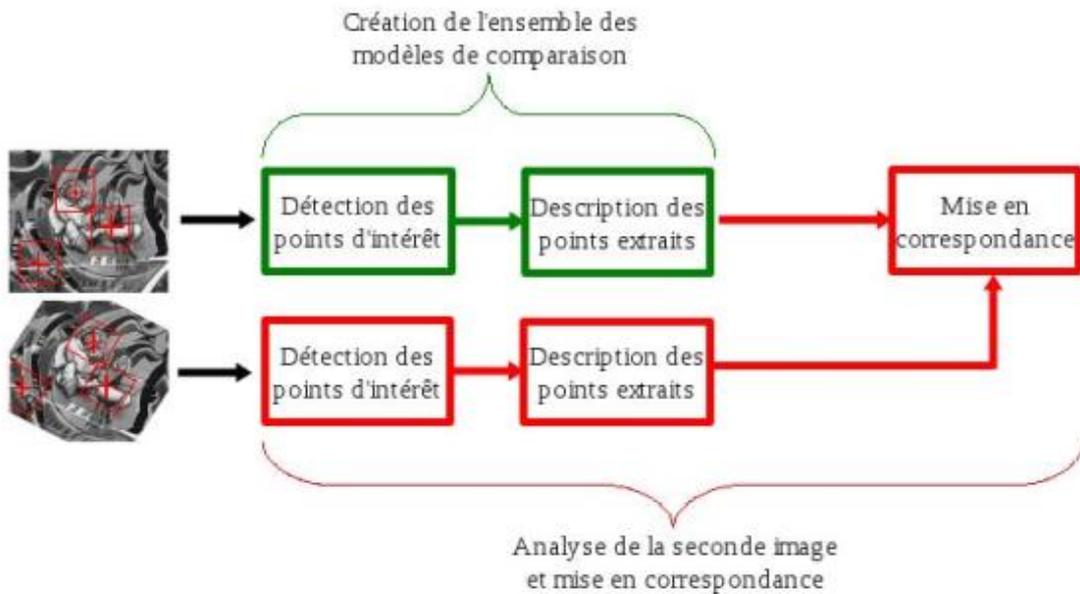


Figure 1. Etapes de l'analyse d'image et de la mise en correspondance.

La première image permet de créer un ensemble de descripteurs caractérisant les points d'intérêt extraits, qui seront par la suite comparés à l'ensemble des descripteurs issus de la seconde image, afin de réaliser des appariements entre ces points.

Problématique et objectifs de cette mémoire :

Le travail de recherche présenté dans ce manuscrit s'appuie essentiellement sur l'analyse d'images, leurs caractérisations et leurs mises en correspondance. Nous avons vu précédemment que de nombreuses applications sont susceptibles d'utiliser un tel procédé. Ce procédé dépendant fortement de la qualité des points d'intérêt et de leurs mises en correspondance.

C'est pourquoi nous avons élaboré une étude comparative de six détecteur et descripteurs des points d'intérêts. Ces descripteurs et détecteurs sont les plus utilisés dans la littérature. Nous avons comparé aussi deux méthodes de mise en correspondance.

Plan du mémoire

Dans ce mémoire nous présentons le traitement de mise en correspondance des images en trois chapitres organisés comme suit :

- Chapitre 1 et 2 : dans les premiers chapitres nous nous présentons le traitement de la correspondance de points d'intérêts, et nous étudions et résumons les méthodes de détection de points d'intérêts et la description des caractéristiques locales en tant que recherche connexe dans chaque traitement.
- Chapitre 3 : Dans ce chapitre nous détaillons notre méthode de test, ainsi que les outils que nous avons utilisés et les résultats que nous avons obtenus, pour faire une comparaison entre un ensemble d'algorithmes de détection et de description des points d'intérêts.

CHAPITRE I :

Détection des points d'intérêts :

La détection des points d'intérêts est le processus de détecter l'existence des points d'intérêts d'une image cible, et c'est le premier processus de mise en correspondance d'images. Ce chapitre explique comment les points d'intérêts sont détectés par les méthodes de détection des coins ainsi que les méthodes invariantes à l'échelle. Les algorithmes mentionnés dans ce chapitre sont : le détecteur d'angle Harris, le détecteur de point d'intérêt par Hessien, Features from Accelerated Segment (FAST), Harris-Laplace et Hessien-Laplace, Scale-Invariant Feature Transform Detector (SIFT), Speeded-Up Robust Features Detector (SURF) et Oriented FAST et Rotated BRIEF (ORB).

1.1. Introduction :

Les points intérêts sont des points uniques qui sont détectés en fonction des informations de gradient et de luminance dans la zone locale de l'image. Si les points correspondants sont obtenus sans détection de point intérêts, les caractéristiques locales sont décrites dans tous les pixels de l'image et la distance est calculée entre un grand nombre de caractéristiques, ce qui nécessite un coût de calcul très élevé. En détectant les points intérêts à l'avance à partir de l'image, la quantité de caractéristique peut être décrite uniquement pour les pixels des points intérêts détectés et la distance peut être calculée, de sorte que le coût de calcul peut être considérablement réduit. De plus, puisque les points intérêts détectent des points caractéristiques dans l'image, il est facile de décrire des caractéristiques locales uniques au niveau des points intérêts, et il est possible d'éviter une dégradation des performances due à une correspondance erronée. Pour la détection de points intérêts, la précision de la position de point intérêts détectée (localisation) et la répétabilité (repeatability) de la détection de point intérêts entre deux images sont importantes, et diverses méthodes ont été proposées. Les détecteurs de coin Moravec [2, 3], les détecteurs Hessian [4], les détecteurs de coin Harris [5], les détecteurs de coin SUSAN [6], etc. ont été proposés comme premières études de détection de points intérêts. Beaucoup sont conçus pour détecter l'intersection des bords dans une image, c'est-à-dire le point d'angle comme point intérêts. De plus, un détecteur de points intérêts qui répond aux changements d'échelle a été proposé en incorporant une analyse de structure d'image basée sur la théorie de l'espace d'échelle [7, 8] dans la détection de points intérêts [1, 9, 10, 11].

1.2. Détection d'angle Harris :

Harris Corner Detector (en anglais) [5] est une méthode pour détecter les points intérêts en définissant les intersections de plusieurs bords (edges) comme des coins (corners). Tout d'abord, les dérivées du premier ordre $I_x(x; y; \sigma_D)$ et $I_y(x; y; \sigma_D)$ de l'image d'entrée I sont calculées dans les directions x et y . La dérivation de l'image est obtenue en convoluant la fonction gaussienne $g(\sigma_D)$ comme indiqué sur la Figure. 1.1 (b) et la Figure. 1.1 (c) dans l'image par la dérivée du premier ordre des directions x et y .

$$I_x(x; y; \sigma_D) = \frac{\partial g(\sigma_D)}{\partial x} * I(x; y) \quad (1.1)$$

$$I_y(x; y; \sigma_D) = \frac{\partial g(\sigma_D)}{\partial y} * I(x; y) \quad (1.2)$$

$$g(\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (1.3)$$

σ_D est le paramètre d'échelle (scale parameter) de la dérivée du premier ordre de la fonction gaussienne et x et y sont des distances à partir de centre du filtre gaussien.

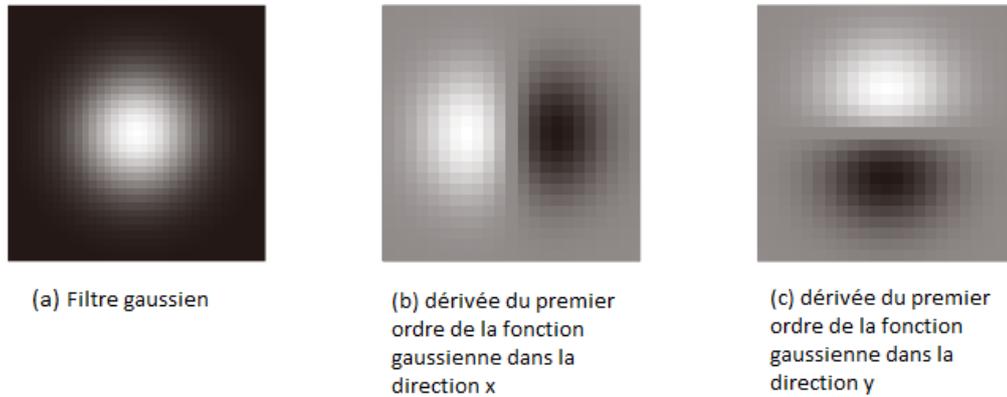


Figure 1. 1. Dérivée première du filtre gaussien

Ensuite, les informations de gradient dans la région locale sont calculées à l'aide de la matrice de moment de second ordre μ indiquée dans l'équation suivante :

$$\mu = \sigma_D^2(\sigma_I) * \begin{bmatrix} I_x^2(x; y; \sigma_D) & I_x(x; y; \sigma_D)I_y(x; y; \sigma_D) \\ I_x(x; y; \sigma_D)I_y(x; y; \sigma_D) & I_y^2(x; y; \sigma_D) \end{bmatrix} \quad (1.4)$$

L'équation (1.4) calcule non seulement la dérivée première de l'image, mais trouve également la somme des dérivées dans la région locale σ_I . Il s'agit de considérer la distribution des valeurs dérivées dans la région locale de l'image, comme le montre la figure 2.2.

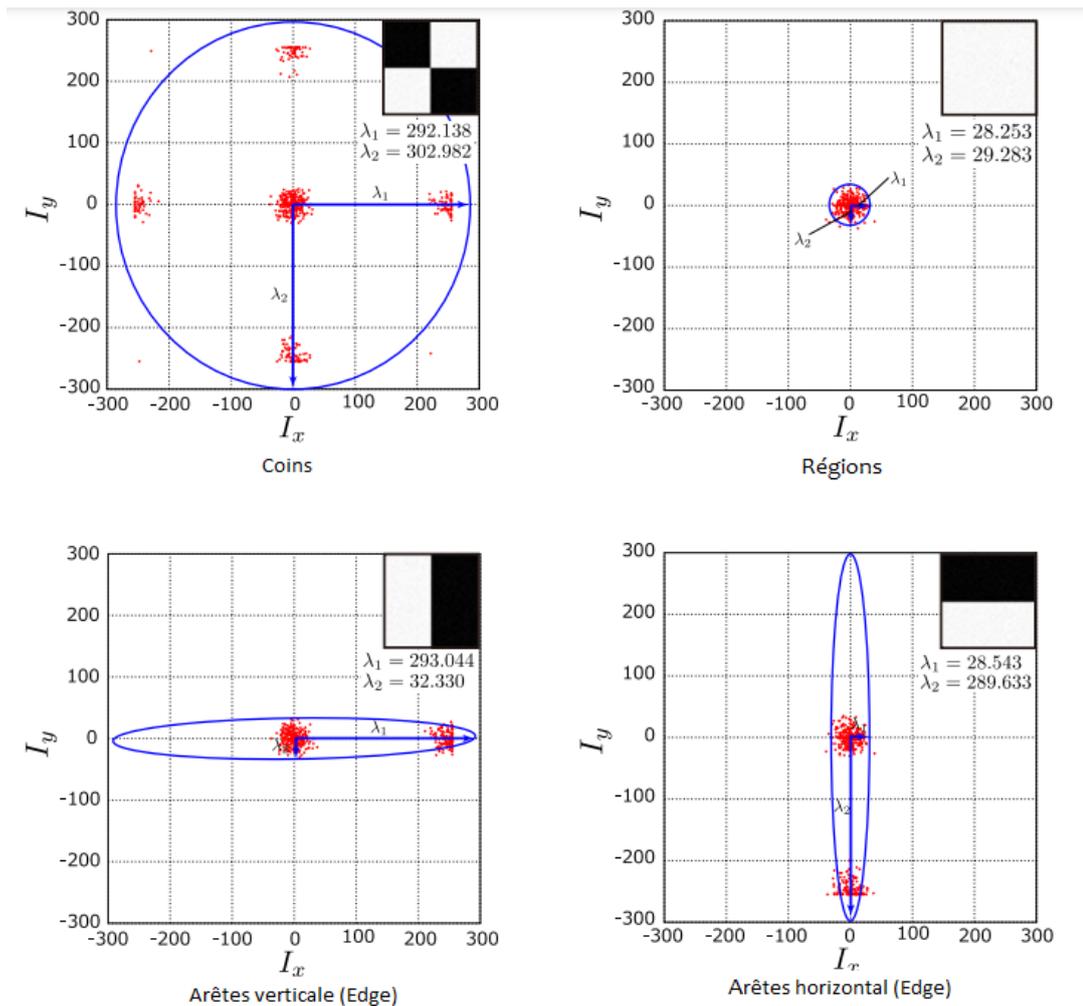


Figure 1.2. Relation entre la distribution des valeurs différentielles dans les coins, les arêtes et les régions plates et les valeurs propres.

Lorsque l'image de la région locale est plate, les valeurs différentielles dans les directions x et y sont petites, de sorte que la variance de la distribution des valeurs différentielles, c'est-à-dire que les première et deuxième valeurs propres de la matrice μ sont petites. Dans le cas de la région de bord, seule la variance de la distribution des valeurs différentielles dans la direction x ou la direction y , c'est-à-dire la première valeur propre de la matrice μ devient grande. Dans le cas de la région d'angle, la variance de la distribution des valeurs différentielles dans chacune des directions x et y devient grande, de sorte que les première et deuxième valeurs propres de la matrice μ deviennent grandes. Afin d'obtenir une telle propriété, l'équation (1.4) trouve la somme des valeurs différentielles dans la région locale. En ce moment, une simple somme des valeurs différentielles de la région locale peut être utilisée, mais une somme pondérée par la fonction gaussienne $g(\sigma_I)$ est souvent utilisée. σ_D est la taille du noyau du filtre dérivé premier gaussien, et σ_I est la taille du noyau pour trouver la somme des valeurs dérivées. Ces deux tailles de noyau sont définies comme $\sigma_D = 0. \sigma_I$ [12].

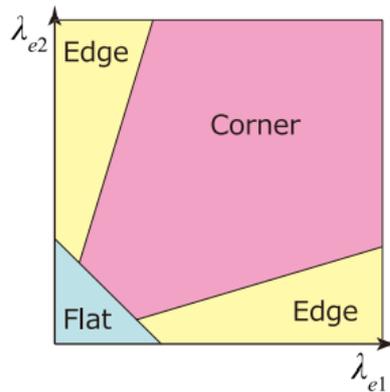


Figure 1.3. Relation entre les valeurs propres de la matrice des moments du second ordre.

À partir de la relation illustrée à la figure 1.3, les valeurs propres λ_{e1} et λ_{e2} de la matrice quadratique μ dans l'équation (1.4) sont liées comme indiqué à la figure 1.3. La méthode de détection de coin basée sur la valeur propre minimale de Shi & Tomasi [26] détecte le coin en seillant la valeur propre minimale $\min(\lambda_{e1}, \lambda_{e2})$ de la matrice μ . Le détecteur de coin de Harris définit le score de coin suivant h_s :

$$h_s = \det(\mu) - k_s \cdot \text{trace}^2(\mu) \quad (1.5)$$

$$\det(\mu) = \lambda_{e1} \cdot \lambda_{e2} \quad (1.6)$$

$$\text{trace}(\mu) = \lambda_{e1} + \lambda_{e2} \quad (1.7)$$

Ainsi, le détecteur de coin de Harris ne résout pas réellement le problème des valeurs propres de la matrice μ , mais calcule le score de coin en combinant le déterminant de la matrice μ et la trace somme des composantes diagonales. k_s est un paramètre d'ajustement du corner score, et $k_s = 0,04$ à $0,06$ est la valeur optimale [27, 28, 29]. Pour les coins détectés, le coin avec le score maximum est laissé dans une petite zone telle que 3×3 pixels, et le processus de suppression non maximum est effectué pour supprimer les autres coins proches avec le score non maximum.

Au moment de la détection de coin, le score de coin h_s est calculé pour chaque pixel de l'image d'entrée, et seul le coin est détecté en fixant une valeur de seuil appropriée pour la valeur h_s . Les positions des sous-pixels des coins peuvent également être calculées en appliquant une fonction quadratique utilisant la valeur de l'équation (1.5) comme dans la méthode d'interpolation de similarité.

1.3. La détection de point intérêts par Hessian :

La détection de point intérêts par Hessian [4] détecte un point où la valeur du pixel d'une image prend une valeur extrême comme point intérêts. L'image peut être vue comme une surface courbe compte tenu du changement continu dans l'espace

tridimensionnel des coordonnées $p = (x, y)$ et de la valeur de pixel $I(p)$. Qu'il s'agisse ou non d'un extremum aux coordonnées p de la surface quadrique peut être déterminé par le déterminant de la matrice Hessienne H montré dans l'équation suivante :

$$H = \sigma_D^2 \begin{bmatrix} I_{xx}(x; y; \sigma_D) & I_{xy}(x; y; \sigma_D) \\ I_{xy}(x; y; \sigma_D) & I_{yy}(x; y; \sigma_D) \end{bmatrix} \quad (1.8)$$

$$I_{xx}(x; y; \sigma_D) = \frac{\partial^2 g(\sigma_D)}{\partial x^2} * I(x; y) \quad (1.9)$$

$$I_{yy}(x; y; \sigma_D) = \frac{\partial^2 g(\sigma_D)}{\partial y^2} * I(x; y) \quad (1.10)$$

$$I_{xy}(x; y; \sigma_D) = \frac{\partial^2 g(\sigma_D)}{\partial xy} * I(x; y) \quad (1.11)$$

σ_D est la taille du noyau du filtre gaussien, et la dérivée seconde de l'image est calculée en convoluant le filtre dans lequel la fonction gaussienne $g(\cdot)$ est secondairement différenciée dans chacune des directions x et y , comme illustré à la figure 1.4. Le jugement suivant peut être fait en trouvant le déterminant de la matrice Hessienne qui a cette valeur en tant qu'élément.

- $det(H) > 0$ et $I_{xx}(x; y; \sigma_D) < 0$: Prendre une valeur maximale à la coordonnée (x, y)
- $det(H) > 0$ et $I_{xx}(x; y; \sigma_D) > 0$: Prend un minimum local à la coordonnée (x, y)
- $det(H) < 0$: Il ne prend pas d'extremum à la coordonnée (x, y) (point selle).
- $det(H) = 0$: On ne sait pas s'il s'agit d'un extremum à la coordonnée (x, y) .

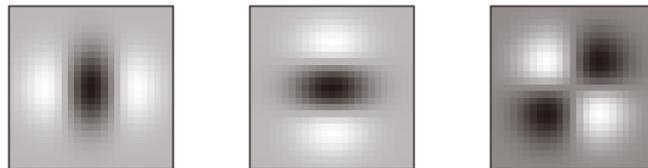


Figure 1.4. Filtre gaussien filtre dérivée seconde

Ce jugement extrême est effectué pour tous les pixels de l'image. La valeur de $det(H)$ est utilisée comme score de point intérêts, et lorsque le score dépasse la valeur de seuil, il est détecté comme point intérêts. Pour les points intérêts détectés, les points intérêts avec le score maximal sont laissés dans une petite zone telle que 3×3 pixels, et un traitement de suppression non maximal est effectué pour supprimer d'autres points intérêts voisins avec des scores non maximaux.

1.4. Features from Accelerated Segment Test (FAST) :

Les détecteurs de coin Harris et les détecteurs Hessiens ont le problème des coûts de calcul élevés car ils doivent calculer la valeur dérivée pour chaque pixel de l'image et convoler le filtre gaussien. Features from Accelerated Segment Test (FAST) [30] apprennent les coins d'une image avec une arbre de décision selon une définition prédéterminée des coins afin de détecter les coins à grande vitesse. L'arbre de décision appris a un mécanisme qui permet une détermination tôt des non-coins, et les coins peuvent être détectés à grande vitesse en recherchant avec l'arbre de décision.

a. Définition des coins :

Le détecteur de coin FAST fait référence à une image sur un cercle concentrique de circonférence de 16 pixels centré sur le pixel d'intérêt p . Ensuite, comme le montre la figure 1.5, lorsque la valeur de luminance sur le cercle concentrique avec un périmètre de 16 pixels est continuellement brillante ou plus sombre que N_{seg} pixels par rapport à la valeur de luminance du pixel d'intérêt, le pixel d'intérêt est défini comme un coin.

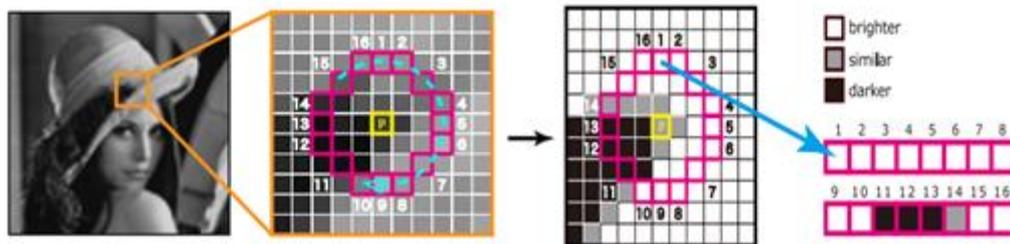


Figure 1.5. Définition des coins dans le détecteur FAST.

Les auteurs rapportent que les expériences montrent la répétabilité la plus élevée lorsque $N_{seg} = 9$ [30]. Lors de la détermination de non-angle avec $N_{seg} = 9$ dans une zone plane, il n'est pas nécessaire d'observer les 16 pixels sur les cercles concentriques, et il est possible de faire une détermination tôt comme non-angle en observant environ 2 pixels. Cependant, selon les points d'observation sur les cercles concentriques, un jugement précoce peut être difficile même dans une région plate. Pour améliorer cela, le détecteur de coin FAST apprend les coins par apprentissage automatique et détermine statistiquement les points d'observation sur les cercles concentriques qui sont les plus susceptibles de séparer les coins et les non-coins. En exprimant les points d'observation déterminés dans une structure arborescente, il est possible de déterminer le coin en recherchant avec l'arbre de décision lorsque le coin est détecté.

b. Construction d'arbres de décision par machine learning :

Lors de la détection de coins selon la définition de coin de FAST, un arbre de décision est construit par apprentissage automatique afin d'observer efficacement les pixels sur des cercles concentriques. Pour apprendre l'arbre de décision, tout d'abord,

pour tous les pixels p de l'image d'entraînement, les pixels sur les cercles concentriques sont classés en trois valeurs : brighter (plus lumineux), similar (similaires) et darker (plus sombres).

$$S(p_c) = \begin{cases} \text{brighter} & I(p) + T_f \leq I(p_c) \\ \text{similar} & I(p) - T_f < I(p_c) < I(p) + T_f \\ \text{darker} & I(p_c) \leq I(p) - T_f \end{cases} \quad (2.12)$$

Ici, $I(p)$ est la valeur de luminance aux coordonnées $p = (x; y)$, $I(p_c)$, $c = \{1, 2, \dots, 16\}$ est la valeur de luminance sur le cercle concentrique de périmètre 16 pixels, et T_f est classé en trois valeurs, il s'agit d'une valeur seuil lors de l'exécution. Comme le montre la figure 1.5, un périmètre ternaire de 16 pixels est généré en tant que vecteur de caractéristiques. Ensuite, si 16 pixels sur le cercle concentrique sont N_{seg} pixels ou plus consécutivement plus clairs ou plus sombres, les coordonnées p sont étiquetées comme des coins, sinon ce sont des non-coins. Ensuite, l'arbre de décision est appris selon l'algorithme de construction d'arbre de décision basé sur ID3 [31] en utilisant le vecteur caractéristique de 16 pixels sur le cercle concentrique ternaire et l'étiquette d'enseignant de la coordonnée p . Au nœud de branche de l'arbre de décision, la valeur $S(p_c)$ sur le cercle concentrique est observée, et le pixel p_c sur le cercle concentrique avec le gain d'information le plus élevé G_{info} obtenu par l'équation (1.13) est déterminé.

$$G_{info} = H_e(P) - H_e(P_b) - H_e(P_s) - H_e(P_d) \quad (1.13)$$

Ici, P est l'ensemble des p qui atteignent le nœud de branche, P_b est l'ensemble des p qui sont déterminés comme étant $S(p_c) = \text{brighter}$, P_s est l'ensemble des p qui sont déterminés comme étant $S(p_c) = \text{similar}$, et P_d est l'ensemble de p qui sont déterminés comme étant $S(p_c) = \text{darker}$. H_e représente l'entropie et peut être calculé à partir de l'équation suivante :

$$H_e(P) = (C + \bar{C}) \log_2(C + \bar{C}) - C * \log_2 C - \bar{C} * \log_2 \bar{C} \quad (1.14)$$

Où C est le nombre d'étiquettes dans les coins qui atteignent chaque nœud de branche, \bar{C} est le nombre d'étiquettes non-coin qui atteint chaque nœud de branche. Dans $H_e(P_b)$, $H_e(P_s)$ et $H_e(P_d)$, l'entropie est calculée en utilisant le nombre d'étiquettes atteintes à chaque nœud de branche, comme dans $H_e(P)$. Ce processus branche les nœuds de l'arbre de décision jusqu'à ce que les coins et les non-coins soient complètement classés, c'est-à-dire jusqu'à ce que le gain d'information devienne nul. Lorsque le gain d'informations devient 0, le nœud est défini comme nœud terminal, l'étiquette atteinte est enregistrée et l'étiquette de sortie finale du nœud terminal est utilisée. La figure 1.6 est un exemple de sélection de pixels sur des cercles concentriques par gain d'information. Lorsque le pixel observé est égale à 2 ,

les coins et les non-coins de l'échantillon d'apprentissage peuvent être classés le plus et $c = 2$ est sélectionné car le gain d'informations est élevé.

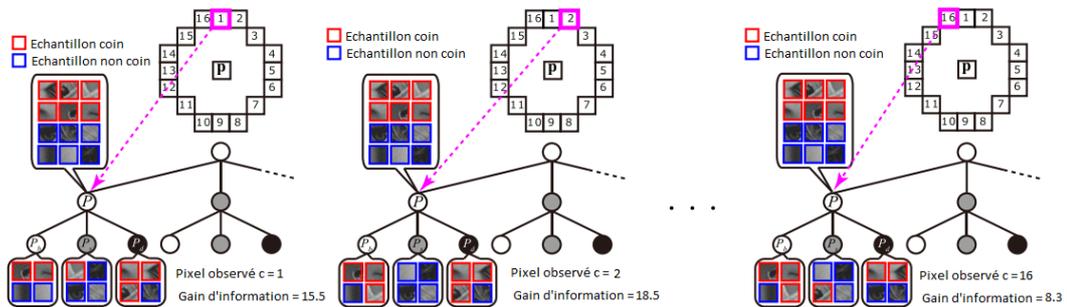


Figure 1.6. Sélection de pixels sur cercles concentriques par gain d'information.

c. Détection des coins par arbre de décision :

Lors de la détection d'un coin par un arbre de décision, les valeurs de luminance des coordonnées p et p_c sont entrées dans l'arbre de décision appris, comme illustré à la figure 1.7 pour se ramifier. Ensuite, le coin ou le non-coin est déterminé sur la base des informations d'étiquette enregistrées dans le nœud terminal atteint. En utilisant un arbre de décision, les pixels sur des cercles concentriques sont efficacement référencés, de sorte que la plupart des pixels de l'image d'entrée peuvent être déterminés tôt. En effet, la plupart des zones locales de l'image sont des zones planes, il est donc possible de déterminer rapidement les non-coins simplement en référençant 2.25 pixels en moyenne pour les pixels sur des cercles concentriques.

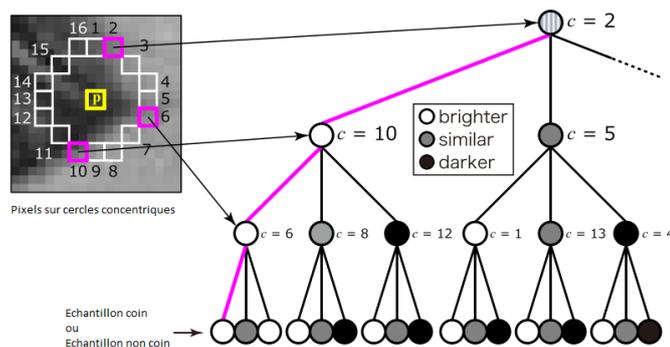


Figure 1.7. Détection des coins par arbre de décision.

d. Suppression non maximale (Non-maximum suppression)

Étant donné que le détecteur de coin FAST détecte les coins en recherchant avec un arbre de décision, le score de coin ne peut pas être obtenu. Le détecteur de coin FAST a également un non-maximum dans une petite zone, similaire à Harris et Hessian. Afin d'effectuer la suppression, il est nécessaire de calculer le score séparément pour le coin détecté. Comme le montre l'équation (1.15), le score dans le détecteur d'angle FAST est calculé comme la somme des valeurs absolues des

différences de luminance entre les pixels des points d'angle et les pixels les plus clairs ou les plus sombres sur le cercle concentrique comme le score f_s .

$$f_s = \max \left(\sum_{c \in S_b} |I(p_c) - I(p)| - T_f, \sum_{c \in S_d} |I(p) - I(p_c)| - T_f \right) \quad (1.15)$$

$$S_b = \{c \mid I(p_c) \geq I(p) + T_f\} \quad (1.16)$$

$$S_d = \{c \mid I(p_c) \leq I(p) - T_f\} \quad (1.17)$$

Ce score est utilisé pour le traitement de suppression non maximale dans une petite zone.

1.5. Détection de points clé à l'aide d'espace d'échelle :

Les méthodes de détection des points intérêts décrite à les sections 1.2, 1.3 et 1.4 ne produisent que les coordonnées des points intérêts. Dans le cas de la sortie des seules coordonnées des points intérêts, une mise en correspondance robuste des points intérêts est possible pour la rotation et la translation de l'image, mais il n'est pas possible de répondre aux changements d'échelle tels que l'agrandissement/la réduction. Nous décrivons ici une méthode de détection de points intérêts invariante à l'échelle qui introduit la théorie de l'espace d'échelle [7, 8, 32, 33, 34] pour la détection de points intérêts.

1.5.1. Harris-Laplace et Hessian-Laplace :

Harris-Laplace et Hessian-Laplace [10] détectent les points intérêts invariants d'échelle en introduisant des espaces d'échelle gaussienne dans les détecteurs de coins de Harris et les détecteurs de Hessian. L'espace d'échelle dans Harris-Laplace et Hessian-Laplace modifie progressivement l'échelle σ_D du filtre gaussien pour donner les scores des équations (1.4) et (1.8). Ensuite, comme le montre la figure 1.8, à chaque niveau de l'espace d'échelle, un point intérêt est détecté lorsque le score du pixel d'intérêt atteint un extremum proche de 26 dans l'espace tridimensionnel de $[x, y, \sigma_D]$. Lors de la détection d'un point intérêt, l'échelle σ_D du pixel d'intérêt est utilisée comme échelle du point intérêt et il est modifier tel que $\{\sigma_{D_1}, \sigma_{D_2}, \dots, \sigma_{D_n}\} = \{k_{sc}\sigma_{D_0}, k_{sc}^2\sigma_{D_0}, \dots, k_{sc}^n\sigma_{D_0}\}$, noter que σ_{D_0} est l'échelle initiale et k_{sc} est le taux d'augmentation de l'échelle. $\sigma_{D_0} = 1.0$ et $k_{sc} = 1.4$, sont fixées respectivement [10].

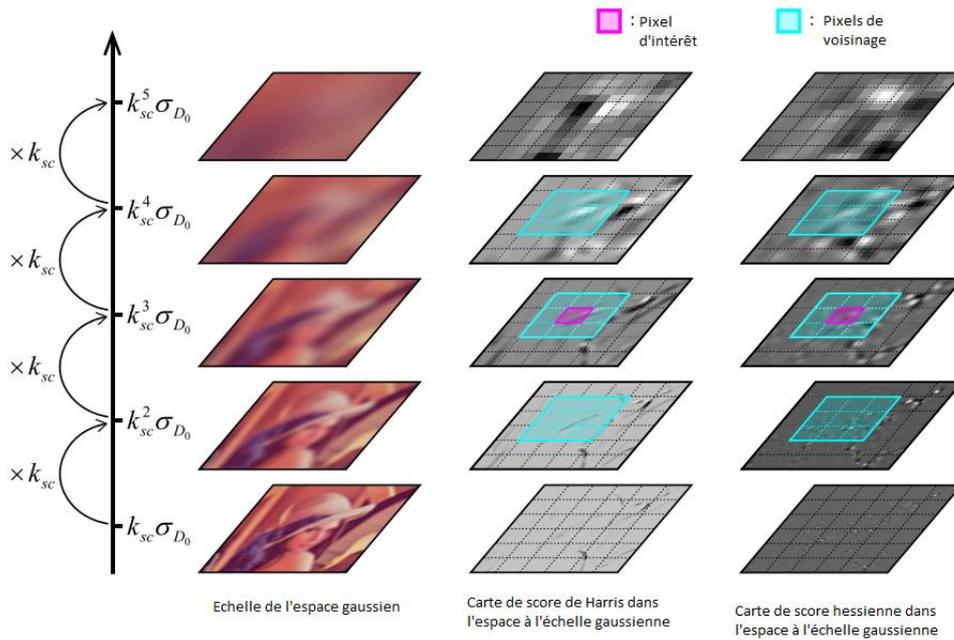


Figure 1.8. Scores de Harris et de Hessian dans l'espace d'échelle

1.5.2. Scale-Invariant Feature Transform (SIFT) Detector :

Scale-Invariant Feature Transform (SIFT) [1] peut détecter des points intéréts qui sont invariants à la rotation de l'image et au changement d'échelle en utilisant un espace d'échelle comme Harris-Laplace et Hessian-Laplace. L'algorithme SIFT comprend deux processus, la détection de points intéréts et la description des caractéristiques, et nous expliquons ici la détection de points intéréts SIFT.

a. Espace d'échelle par différence de gaussienne (DoG) :

La détection des points intéréts par DoG est obtenue à partir de la différence de l'image lissée $L(p; \sigma)$ convoluant la fonction gaussienne $g(\sigma)$ d'échelle différente et l'image d'entrée $I(p)$.

$$L(p; \sigma) = g(\sigma) * I(p) \quad (1.18)$$

Si l'image DoG est $D(p; \sigma)$, l'image DoG peut être calculée par l'équation suivante :

$$\begin{aligned} D(p; \sigma) &= (g(k_{sc}\sigma) - g(\sigma)) * I(p) \\ &= L(p; k_{sc}\sigma) - L(p; \sigma) \end{aligned} \quad (1.19)$$

Ce processus est effectué entre différentes échelles qui sont k_{sc} fois plus grandes que l'échelle initiale σ_0 , et plusieurs images DoG sont obtenues. Si σ continue d'augmenter à un rythme constant, la taille du filtre gaussien augmentera et pourra

être traitée. Le problème se pose qu'il n'y a pas de zone de bordure de l'image et le coût de calcul augmente. Pour résoudre ce problème, nous effectuons un traitement de lissage qui maintient la continuité des changements de σ en sous-échantillonnant l'image.

b. Processus de lissage efficace qui maintient la continuité de σ :

Dans le processus de lissage efficace qui maintient la continuité de σ , l'image d'entrée est d'abord lissée avec la valeur initiale σ_0 , et l'image lissée $L_1(p; \sigma_0)$ est acquise. Ensuite, l'image est lissée avec $k_{sc}\sigma_0$, qui est la valeur obtenue en multipliant σ_0 par k_{sc} , et $L_1(p; k_{sc}\sigma_0)$ est généré. Par le même traitement, plusieurs images lissées avec différents σ sont générées. L'ensemble de traitement jusqu'à ce point est appelé 1 octave. Ensuite, l'image $L_1(p; 2\sigma_0)$ lissée de $2\sigma_0$ est sous-échantillonnée à une taille de 1/2 à partir des multiples images lissées générées. Le nombre de processus de lissage dans une octave sera décrit ultérieurement avec le réglage du taux d'augmentation k_{sc} . La relation suivante est valable entre l'image $L_2(p; 2\sigma_0)$ sous-échantillonnée à une taille de 1/2 et l'image $L_1(p; 2\sigma_0)$ lissée de $2\sigma_0$.

$$L_1(p; 2\sigma_0) \approx L_1(p; \sigma_0) \tag{1.20}$$

En utilisant cette relation, la plage de σ peut être limitée et l'augmentation de la quantité de calcul due à la taille du filtre gaussien peut être empêchée.

c. Recherche de valeur extrême pour les images DoG :

Étant donné que DoG est la différence entre des images lissées à différentes échelles, σ avec un grand score DoG contient une grande quantité d'informations telles que les bords dans la zone où l'échelle change. Par conséquent, l'extremum est détecté à partir de l'image DoG, et le candidat de point intérêts et son échelle sont déterminés. Les valeurs extrêmes sont détectées à partir d'un ensemble de trois images DoG, comme le montre la figure 1.9. Lorsque le pixel d'intérêt dans l'image DoG entouré par la ligne pointillée rouge de la figure 1.9 est comparé à la valeur voisine de 26 dans l'espace tridimensionnel de $[x, y, \sigma]$, et que le pixel d'intérêt est le extremum, ce pixel est détecté comme candidat point intérêts. De cette manière, il est possible de détecter des candidats de points intérêts qui prennent en compte à la fois l'espace $[x, y]$ et l'espace d'échelle.

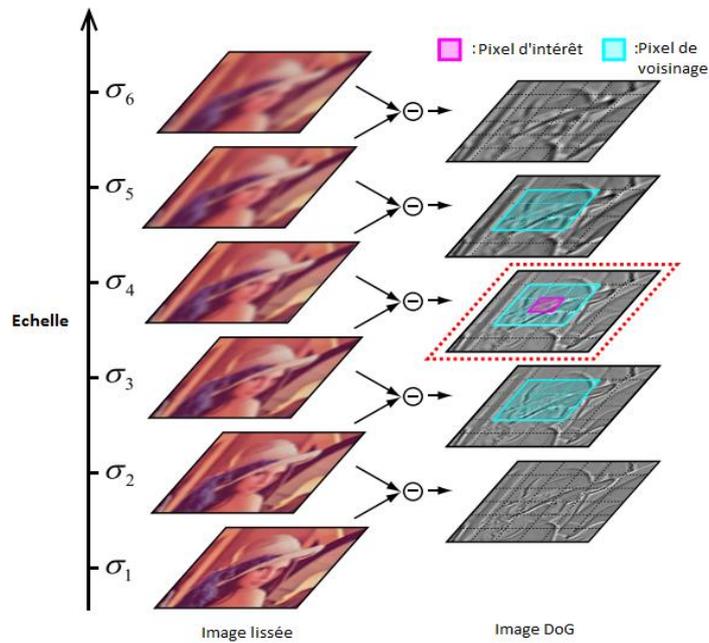


Figure 1.9 . Détection extrême à partir d'images DoG

La détection de valeur extrême est effectuée à partir d'une image DoG avec un petit σ , et les pixels pour lesquels une valeur extrême a été détectée une fois ne sont pas recherchés pour les valeurs extrêmes sur les grandes échelles suivantes. Ce processus est effectué sur toutes les images DoG de différentes échelles.

d. Supprimer les points intérêts candidats sur le bord :

Les points intérêts candidats détectés par la recherche de valeurs extrêmes de l'image DoG incluent les points intérêts candidats détectés sur le bord de l'image, et sont facilement affectés par le problème d'ouverture lors de la mise en correspondance des points intérêts. Par conséquent, les points intérêts candidats existant sur le bord sont supprimés des points intérêts candidats.

Tout d'abord, la matrice Hessienne bidimensionnelle H_{DoG} dans le point intérêts candidat est calculée par l'équation suivante :

$$H_{DoG} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{1.21}$$

La dérivée dans la matrice est obtenue à partir de la dérivée seconde de la valeur de sortie DoG à la position candidate du point intérêts. Ici, la première valeur propre obtenue à partir de la matrice Hessienne H_{DoG} est λ_α , et la deuxième valeur propre est λ_β ($\lambda_\alpha > \lambda_\beta$).

A ce moment, la *trace* (H_{DoG}) des composantes diagonales de la matrice hessienne et le déterminant $det(H_{DoG})$ peuvent être calculés comme suit :

$$trace(H_{DoG}) = D_{xx} + D_{yy} = \lambda_\alpha + \lambda_\beta \tag{1.22}$$

$$\det(H_{DoG}) = D_{xx}D_{yy} - D_{yy}^2 = \lambda_\alpha \lambda_\beta \quad (1.23)$$

Si le taux de la première valeur propre et de la deuxième valeur propre est γ et $\lambda_\alpha = \gamma \lambda_\beta$, l'équation suivante est obtenue :

$$\frac{\text{trace}^2(H_{DoG})}{\det(H_{DoG})} = \frac{(\lambda_\alpha + \lambda_\beta)^2}{\lambda_\alpha \lambda_\beta} = \frac{(\gamma \lambda_\beta + \lambda_\beta)^2}{\gamma \lambda_\beta^2} = \frac{(\gamma + 1)^2}{\gamma} \quad (1.24)$$

En supposant que le seuil du rapport de $\text{trace}^2(H_{DoG})$ et $\det(H_{DoG})$ est γ^{emme} , le rapport de la première valeur propre et de la deuxième valeur propre lorsque le rapport de $\text{trace}^2(H_{DoG})$ et $\det(H_{DoG})$ est inférieur au seuil comme indiqué dans l'équation suivante est déterminé à être petit et reste comme un candidat intérêts.

$$\frac{\text{trace}^2(H_{DoG})}{\det(H_{DoG})} < \frac{(\gamma^{emme} + 1)^2}{\gamma^{emme}} \quad (1.25)$$

Ce processus est similaire à la détermination des coins du détecteur de coins de Harris décrit dans la section 1.2, et il n'est pas nécessaire de résoudre réellement le problème des valeurs propres de la matrice H_{DoG} . Dans la référence [1], $\gamma^{emme} = 10$ est adopté, et le membre de droite de l'équation (1.25) est 12,1.

e. Estimation de la position des sous-pixels du point intérêts:

La valeur de sortie de DoG peut être corrigée par l'ajustement de la fonction quadratique des trois variables $[x, y, \sigma]$ pour corriger la position et l'échelle du sous-pixel du point intérêts. Les coordonnées du point intérêts et échelle de la fonction DoG $D(q)$ à $q = [x, y, \sigma]^T$. L'approximation avec un développement quadratique de Taylor donne $D_{apx}(q)$ comme indiqué dans l'équation suivante.

$$D_{apx}(q) \approx D + \frac{\partial D^T}{\partial q} q + \frac{1}{2} q^T \frac{\partial^2 D}{\partial q^2} q = D + \frac{\partial D^T}{\partial q} q + \frac{1}{2} \frac{\partial^2 D}{\partial q^2} q^2 \quad (1.26)$$

Dans l'équation (1.26), $\hat{q} = [\hat{x}, \hat{y}, \hat{\sigma}]^T$, tel que la dérivée partielle par rapport à q soit 0 soit l'extremum à la position et à l'échelle exactes.

$$\frac{\partial D_{apx}}{\partial q} = \frac{\partial D}{\partial q} + \frac{\partial^2 D}{\partial q^2} q = 0 \quad (1.28)$$

$$\frac{\partial^2 D}{\partial q^2} q = - \frac{\partial D}{\partial q} \quad (1.29)$$

$$\hat{q} = - \frac{\partial^2 D^{-1}}{\partial q^2} \frac{\partial D}{\partial q} \quad (1.30)$$

A ce moment, $\hat{q} = [\hat{x}, \hat{y}, \hat{\sigma}]^T$ représente la position du sous-pixel, et lorsque l'équation (1.30) est exprimée sous forme de matrice, elle devient l'équation suivante.

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\sigma} \end{bmatrix} = - \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial x\sigma} \\ \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y\sigma} \\ \frac{\partial^2 D}{\partial x\sigma} & \frac{\partial^2 D}{\partial y\sigma} & \frac{\partial^2 D}{\partial \sigma^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial \sigma} \end{bmatrix} \quad (1.31)$$

En résolvant l'équation (1.31), les sous-pixels des points intérêts candidats et l'échelle précise peuvent être estimés. Par conséquent, l'estimation de position de sous-pixel est efficace non seulement pour la correction de position mais également pour la correction d'échelle.

f. Supprimer les points intérêts candidats à faible contraste :

Puisque la recherche de valeurs extrêmes capture une petite valeur extrême, de nombreux points intérêts avec un faible score DoG (low contrast) sont inclus dans les points intérêts candidats. Étant donné que les points intérêts à faible contraste sont sensibles au bruit, ces points intérêts sont supprimés. Le score DoG $D(\hat{q})$ à la position du sous-pixel peut être calculé comme suit en remplaçant l'équation (1.30) par l'équation (1.26).

$$\begin{aligned} D(\hat{q}) &= D + \frac{\partial D^T}{\partial q} \hat{q} + \frac{1}{2} \hat{q}^T \frac{\partial^2 D}{\partial q^2} \hat{q} \quad (1.32) \\ &= D + \frac{\partial D^T}{\partial q} \hat{q} + \frac{1}{2} \left(-\frac{\partial^2 D^{-1}}{\partial q^2} \frac{\partial D}{\partial q} \right)^T \frac{\partial^2 D}{\partial q^2} \hat{q} \\ &= D + \frac{\partial D^T}{\partial q} \hat{q} - \frac{1}{2} \frac{\partial D^T}{\partial q} \hat{q} \\ &= D + \frac{1}{2} \frac{\partial D^T}{\partial q} \hat{q} \end{aligned}$$

Le score DoG à la position du sous-pixel peut être calculé par l'équation (1.32). Après avoir normalisé la valeur absolue du score dans le sous-pixel à $|D(\hat{q})| \in [0, 1]$, les points intérêts à faible contraste, c'est-à-dire les points intérêts à faible score DoG, sont supprimés en traitant avec le seuil valeur. Dans la référence [1], le seuil de suppression des points intérêts à faible contraste est fixé à 0,03.

g. Calcul d'orientation :

SIFT calcule l'orientation, qui est la direction principale à chaque position de point intéréts, afin de décrire les caractéristiques qui sont invariantes à la rotation de l'image. Pour calculer l'orientation, comme le montre la Figure 1.10, à partir de l'image lissée $L(p; \hat{\sigma})$ centrée sur le point intéréts, l'intensité du gradient $m(p)$ et la direction du gradient $o(p)$ sont la plage de l'échelle du point intéréts $\hat{\sigma}$. Obtenu à partir de :

$$m(p) = \sqrt{g_x(p)^2 + g_y(p)^2} \tag{1.33}$$

$$o(p) = \tan^{-1}\left(\frac{g_y(p)}{g_x(p)}\right) \tag{1.34}$$

$$\begin{cases} g_x(p) = L(x + 1, y; \hat{\sigma}) - L(x - 1, y; \hat{\sigma}) \\ g_y(p) = L(x, y + 1; \hat{\sigma}) - L(x, y - 1; \hat{\sigma}) \end{cases} \tag{1.35}$$

A partir de l'intensité du gradient $m(p)$ et de la direction du gradient $o(p)$ dans la région d'échelle, l'histogramme de gradient pondéré $hist(o')$ est obtenu à partir de l'équation suivante.

$$hist(o') = \sum_x \sum_y g(\hat{\sigma}) * m(p) . \delta[o', o(p)] \tag{1.36}$$

$hist(o')$ est un histogramme obtenu en quantifiant la direction du gradient dans 36 directions, et vote pour l'intensité du gradient pondérée par la fonction gaussienne $g(\hat{\sigma})$ de l'échelle de point intéréts $\hat{\sigma}$. La pondération par la fonction gaussienne donne un poids important à la force du gradient près du point intéréts. $\delta[\cdot]$ est une fonction delta de Kronecker qui renvoie 1 lorsque la direction du gradient $o(p)$ est quantifiée et correspond à la direction du gradient de quantification o' .

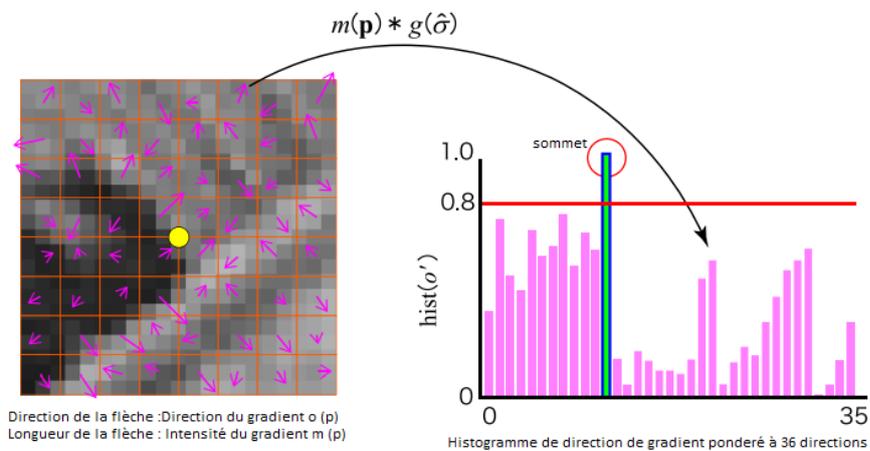


Figure 1. 2. Calcul de l'orientation SIFT.

Tous les casiers dans la direction du gradient qui sont à 80 % ou plus de la valeur maximale de cet histogramme pondéré de la direction du gradient sont affectés en tant qu'orientation $\hat{\theta}$ du point intérêts. Par conséquent, les points intérêts détectés à partir de positions telles que des coins se voient attribuer des orientations dans deux directions ou plus. Lors de leur description, les caractéristiques sont décrites dans chaque direction. De plus, SIFT calcule l'orientation comme une valeur continue en appliquant l'ajustement polynomial de la fonction quadratique à l'histogramme de direction du gradient. Ce processus permet un calcul précis de l'orientation.

1.5.3. Speeded-Up Robust Features (SURF) Detector :

Les fonctionnalités robustes accélérées (SURF) [9] peuvent également détecter des points intérêts qui sont invariants à la rotation de l'image et aux changements d'échelle, et comme SIFT, il consiste en un algorithme en deux étapes de détection des points intérêts et de description des caractéristiques. Ici, nous expliquons la détection de point intérêts de SURF.

a. Approximation de Hessian par filtre Box :

La détection de points intérêts par SURF détecte les points intérêts sur la base du traitement Hessian-Laplace. Cependant, dans le Hessian-Laplace, le coût de calcul est élevé car l'extremum est détecté en faisant convoluer le filtre gaussien de dérivée seconde dans l'espace d'échelle. Par conséquent, SURF accélère le processus de détection des points intérêts en rapprochant le filtre gaussien de la dérivée seconde avec un simple filtre Box. En combinant le filtre Box avec l'image intégrée, il est possible de filtrer à grande vitesse. Comme le montre la figure 1.11, les filtres gaussiens de la dérivée seconde g_{xx} , g_{yy} et g_{xy} peuvent être approchés par les filtres Box B_{xx} , B_{yy} et B_{xy} .

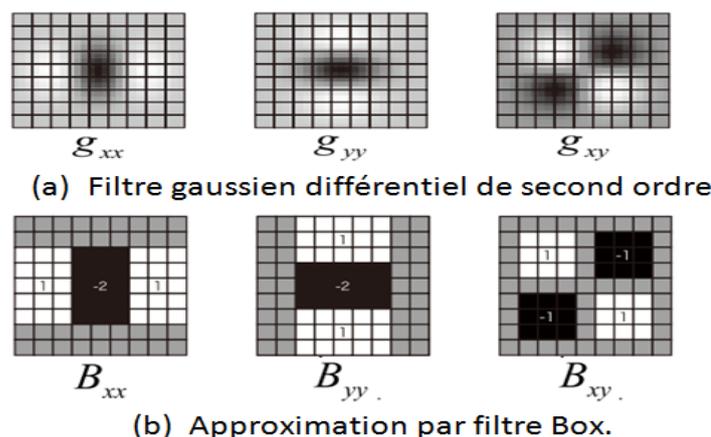


Figure 1. 3. Approximation du filtre Gaussien en dérivée seconde par le filtre Box

Lorsque la dérivée seconde de l'image calculée par le filtre Box est respectivement $I_{xx}(p, \sigma)$, $I_{yy}(p, \sigma)$ et $I_{xy}(p, \sigma)$, la matrice hessienne approximative H_{apx} peut être calculée comme suit.

$$H_{apx} = \begin{bmatrix} I_{xx}(p, \sigma) & I_{xy}(p, \sigma) \\ I_{xy}(p, \sigma) & I_{yy}(p, \sigma) \end{bmatrix} \quad (1.37)$$

$$I_{xx}(p, \sigma) = B_{xx}(\sigma) * I(p) \approx g_{xx}(\sigma) * I(p) \quad (1.38)$$

$$I_{yy}(p, \sigma) = B_{yy}(\sigma) * I(p) \approx g_{yy}(\sigma) * I(p) \quad (1.39)$$

$$I_{xy}(p, \sigma) = B_{xy}(\sigma) * I(p) \approx g_{xy}(\sigma) * I(p) \quad (1.40)$$

L'échelle σ dans le filtre Box modifie la taille du filtre en fonction de σ . Par exemple, si l'échelle du filtre gaussien passe à $= \{1,2; 2,0; 2,8; 3,6\}$, la taille du filtre Box passe à $\{9 \times 9; 15 \times 15; 21 \times 21; 27 \times 27\}$ et ainsi de suite.

Semblable à la détection de points intérêts par Hessian, le score de points intérêts est obtenu en calculant le déterminant de la matrice Hessienne approximative $\det(H_{apx})$

$$\det(H_{apx}) = I_{xx}(p, \sigma) \cdot I_{yy}(p, \sigma) - (0,9 \cdot I_{xy}(p, \sigma))^2 \quad (1.41)$$

Ici, $I_{xy}(p, \sigma)$ est pondéré par 0,9, qui est un facteur de pondération relative pour équilibrer le déterminant $\det(H_{apx})$ de manière bien équilibrée. Si les images de dérivée seconde calculées par les filtres gaussiens de dérivée seconde g_{xx} et g_{xy} sont I_{xx} et I_{xy} , la relation de la norme de Frobenius de l'image de dérivée seconde comme montré dans l'équation suivante peut être obtenue.

$$\frac{\|I_{xy}(\cdot; 1,2)\|_F \cdot \|I_{xx}(\cdot; 9)\|_F}{\|I_{xx}(\cdot; 1,2)\|_F \cdot \|I_{xy}(\cdot; 9)\|_F} = 0,912 \dots \approx 0,9 \quad (1.42)$$

Le résultat du calcul du filtre est normalisé par rapport à la taille, et une norme de Frobenius constante est garantie pour toute taille de filtre, de sorte que le facteur de pondération est fixé à 0,9.

b. Recherche de valeurs extrêmes par espace d'échelle du filtre Box :

L'estimation d'échelle dans SURF calcule le score du point intérêt dans l'espace d'échelle en modifiant la taille du filtre Box. Comme le montre la figure 1.12, le score est calculé dans l'espace d'échelle par le filtre Box, puis l'extremum est recherché en comparant avec les pixels au voisinage de 26 comme dans SIFT.

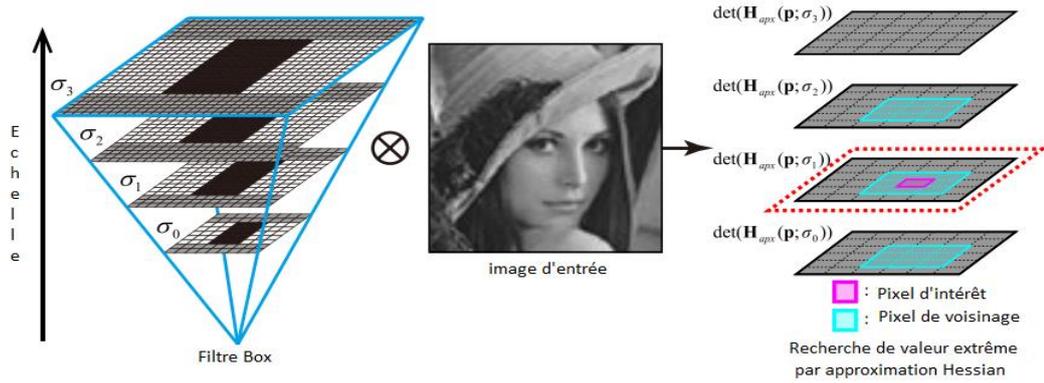


Figure 1. 4. Recherche de valeurs extrêmes à l'aide du filtre Box.

Lorsque le pixel d'intérêt est un extremum, les coordonnées de cette position et la taille du filtre Box sont détectées comme des points intérêts et des échelles.

c. Calcul d'orientation :

Comme le montre la figure. 1.13, l'orientation SURF est calculée en calculant le *Haar – Wavelet* ($4\hat{\sigma} \times 4\hat{\sigma}$) dans les directions x et y à partir de la région de $6\hat{\sigma}$ centrée sur le point intérêt. $\hat{\sigma}$ est l'échelle des points intérêts. La somme est calculée pour chaque direction x et y tout en faisant tourner le gradient de luminosité calculé dans la région de $6\hat{\sigma}$ par $\pi/3$ autour du point intérêts. La direction avec la plus grande somme des gradients de luminosité calculés est adoptée comme orientation du point intérêts $\hat{\theta}$.

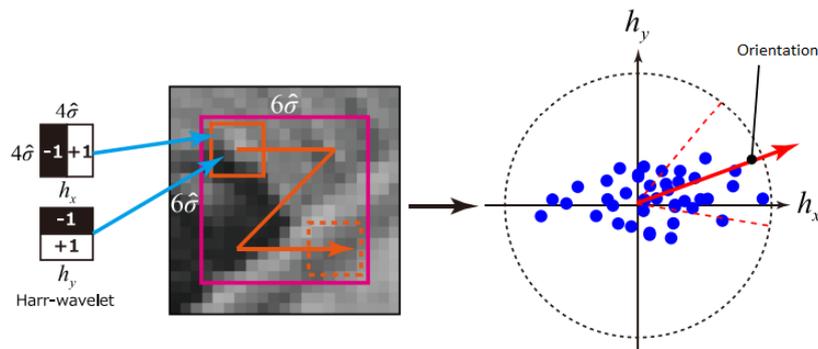


Figure 1. 5. Calcul de l'orientation SURF

1.5.4. Détecteur orienté FAST et rotatif BRIEF (ORB) :

Oriented FAST et Rotated BRIEF (ORB) réalisent une détection de point intérêts à grande vitesse tout en répondant aux changements d'échelle en combinant une pyramide d'image et un détecteur de coin FAST. Nous proposons également une description de caractéristiques binaires à haute vitesse et économe en mémoire basée sur la différence de luminosité de la paire de pixels d'échantillonnage. Ici, nous décrivons la détection de points intérêts dans ORB. L'ORB utilise le détecteur de coin FAST décrit dans la section 1.4 pour détecter les points intérêts à grande vitesse. Bien que le détecteur de coin FAST soit capable d'un traitement à grande vitesse, il ne

calculer pas l'échelle ou l'orientation des points intérêts, il y a donc un problème que l'invariance ne peut pas être obtenue pour les changements d'échelle et la rotation pendant la correspondance des points intérêts. Par conséquent, ORB utilise un détecteur d'angle FAST pour calculer l'échelle et l'orientation.

a. Acquisition d'échelle par pyramide d'images :

Tout d'abord, une pyramide d'images est générée en sous-échantillonnant l'image d'entrée en plusieurs étapes afin d'obtenir une invariance d'échelle. La pyramide d'image est générée en sous-échantillonnant l'image d'entrée de $1/\sqrt{2}$ fois, comme illustré à la figure 1.14. Les coins de l'image générée sont détectés par le détecteur de coins FAST à partir des images de chaque couche de la pyramide. L'inverse du grossissement de l'image dans laquelle le coin est détecté est utilisé tel quel comme l'échelle $\hat{\sigma}$. De plus, le score du détecteur de coin de Harris tel qu'illustré dans l'équation (1.5) est calculé pour les points de coin détectés. Seuls les coins avec ce score au-dessus du seuil sont détectés comme points intérêts.

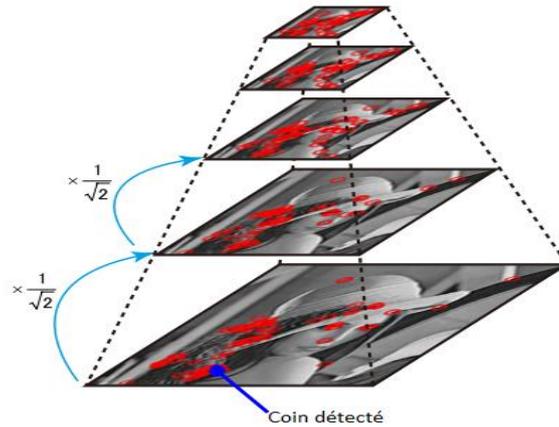


Figure 1. 6. Acquisition d'échelle par pyramide d'images

b. Calcul d'orientation :

Pour calculer l'orientation, le moment 0, 1 ; $\tilde{m}_{uv}(u, v = \{0, 1\})$ est obtenu à partir de la valeur de luminosité de l'image du patch dans la plage d'échelle obtenue par la pyramide d'image.

$$\tilde{m}_{uv} = \sum_{x,y} x^u y^v I(x, y) \tag{1.43}$$

À partir du moment calculé d'après l'équation on obtient la position du centre de gravité C de l'image.

$$C = \left(\frac{\tilde{m}_{10}}{\tilde{m}_{00}}, \frac{\tilde{m}_{01}}{\tilde{m}_{00}} \right) \tag{1.44}$$

Le vecteur de direction de la position du point intérêts à la position du centre de gravité de l'image est l'orientation $\hat{\theta}$, qui peut être simplement calculée par l'équation suivante.

$$\hat{\theta} = \tan^{-1} \left(\frac{\tilde{m}_{01}}{\tilde{m}_{10}} \right) \quad (1.45)$$

1.6. Conclusion :

Dans ce chapitre, les déroulements des traitements de détection des points intérêts proposés jusqu'à présent sont décrits.

Les premières études de détection de points intérêts [2, 4, 5, 6, 26, 27] ont détecté des points intérêts qui n'étaient invariants qu'aux translations et aux changements de rotation entre les images, mais en introduisant l'espace d'échelle de l'image, il est devenu possible de détecter des points intérêts résistants aux changements d'échelle [1, 7, 8, 9, 10]. En termes d'accélération de la détection de points intérêts, la détection de points intérêts à grande vitesse a été obtenue en recherchant efficacement avec un arbre de décision construit par apprentissage automatique [30].

CHAPITRE II

Description des caractéristiques locales et la mise en correspondances des points clé

La description des caractéristiques local est le deuxième processus dans la mise en correspondances. Ce chapitre exprime comment les descriptions sont fait quelque soit avec un vecteur binaire ou réel. Les algorithmes mentionnés dans ce chapitre sont : Scale-Invariant Feature Transform (SIFT) Descriptor , Speeded-Up Robust Features (SURF) Descriptor, PCA-SIFT , Gradient Location and Orientation Histogram (GLOH), Binary Robust Independent Elementary Features (BRIEF), Binary Robust Invariant Scalable Keypoints (BRISK), Oriented FAST and Rotated BRIEF (ORB) Descriptor, Fast Retina Keypoint (FREAK).

2.1. Introduction :

La quantité de caractéristique locale exprime des informations telles que le gradient et la texture de l'image du point clé détecté à partir de l'image en tant que vecteur de caractéristique. Lorsque la valeur de luminosité d'une image est adoptée en tant que vecteur caractéristique telle quelle, il est difficile d'obtenir un point de correspondance précis entre les images car elle est fortement affectée par les changements d'éclairage et le bruit. Par conséquent, dans la description de caractéristique locale, le vecteur de caractéristique qui n'est pas facilement affecté par les changements d'éclairage et le bruit est calculé à l'aide d'informations telles que l'histogramme de direction de gradient et la différence de luminosité entre les pixels. Pour les caractéristiques locales, de nombreuses méthodes [9, 14, 15] ont été proposées pour calculer des vecteurs de caractéristiques réels basés sur les informations de gradient des images locales, y compris Scale-Invariant Feature Transform (SIFT) [1]. Cependant, alors que la précision de la correspondance des points clés est élevée pour les caractéristiques locales utilisant un vecteur réel, il existe des problèmes tels qu'une augmentation de l'utilisation de la mémoire et l'incapacité de décrire les caractéristiques à grande vitesse. Comme solution à ce problème, une méthode d'expression des caractéristiques locales sous la forme d'un vecteur binaire de $\{0, 1\}$ ou $\{-1, 1\}$ a été proposée [16, 17, 18, 19, 20]. La majeure partie de la description des caractéristiques par vecteur binaire est $\{0, 1\}$ ou $\{-1, 1\}$ en fonction de la différence de luminosité entre les pixels de l'image du patch autour du point clé et du signe du résultat du produit de l'image et du filtre linéaire. Puisqu'il est attribué, la description des fonctionnalités à grande vitesse est possible. De plus, étant donné que toutes les caractéristiques sont représentées par deux valeurs, une mise en correspondance des points clés permettant d'économiser de la mémoire est possible. Étant donné que la description de la quantité de caractéristiques à l'aide d'un vecteur binaire réduit la quantité d'informations et que la précision de la correspondance des points clés est inférieure à celle d'un vecteur de nombre réel, la quantité de caractéristiques basée sur le vecteur de nombre réel est convertie en vecteur binaire par la matrice de transformation de caractéristiques. Des études sur la description des caractéristiques de valeur sont également en cours [21, 22, 23, 24].

2.2. Description des caractéristiques local par vecteur réel

Cette section décrit la description des caractéristiques locales dans la correspondance des points clés. La description de caractéristique locale peut être divisée en une méthode d'expression de la caractéristique par un vecteur de nombre réel et une méthode d'expression de la caractéristique par un vecteur binaire. Nous décrivons ici une méthode pour exprimer des caractéristiques locales par un vecteur réel.

2.2.1. Scale-Invariant Feature Transform (SIFT) Descriptor :

La transformation de caractéristiques invariantes à l'échelle (SIFT) se compose de deux processus, la détection de points clés et la description de caractéristiques locales, et la méthode de détection de points clés est décrite à la section 1.5.2. Ici, la description des caractéristiques locales, qui est le traitement ultérieur de SIFT, sera expliquée.

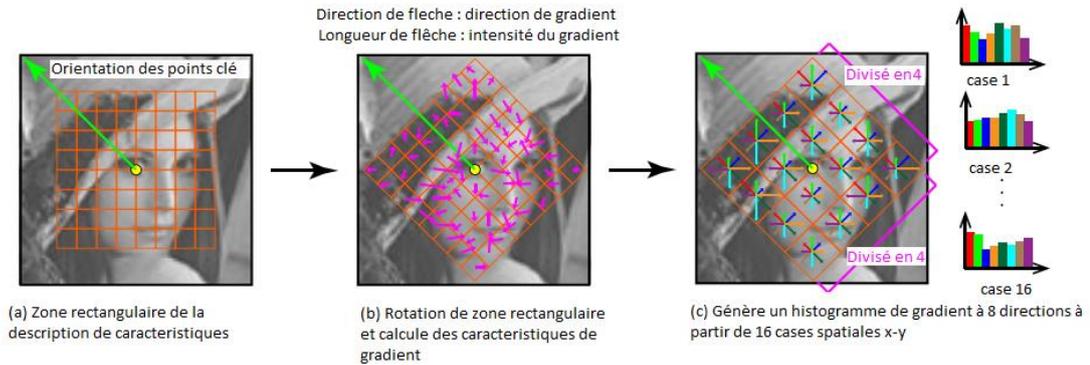


Figure 2.1. Description des caractéristiques SIFT.

La quantité de caractéristiques basée sur le gradient est calculée à l'aide de la position du point clé p détectée à partir de l'image, de l'échelle $\hat{\sigma}$ et de l'orientation $\hat{\theta}$. Tout d'abord, comme le montre la figure. 2.1 (a), la zone décrivant les caractéristiques est tournée dans la direction d'orientation des points clés. Les informations de gradient obtenues à partir de la zone rectangulaire inscrite par l'échelle du point clé sont utilisées pour décrire les caractéristiques. L'intensité du gradient et la direction du gradient sont calculées à partir des cases dans l'espace x - y constitué d'un total de 16 blocs dans lesquels un côté de la zone rectangulaire est divisé en 4 blocs. L'intensité et la direction du gradient sont calculées par les équations (1.33) et (1.34) définies dans la détection de point de repère SIFT, respectivement. Comme le montre la figure. 2.1 (c), un histogramme de direction du gradient dans 8 directions (par incréments de 45 °) est généré à partir des cases dans chaque espace x - y . Cet histogramme de gradient est identique à l'histogramme de gradient créé lors du calcul de l'orientation SIFT. Afin de calculer l'histogramme de direction du gradient dans 8 directions à partir du case « 4 × 4 » ; $4 \times 4 \times 8 = 128$ caractéristiques dimensionnelles sont finalement décrites. En décrivant les caractéristiques locales en fonction de l'échelle et de l'orientation des points clés de cette manière, les caractéristiques sont invariantes aux changements d'agrandissement / réduction et de rotation.

Enfin, afin de réduire l'influence des changements d'éclairage entre les images, les caractéristiques de la $i^{\text{ème}}$ dimension sont normalisées par unité de longueur, comme indiqué dans l'équation suivante.

$$d_i = \frac{d_i}{\sum_{j=1}^{N_{dim}} |d_j|} \quad (2.1)$$

Ici, N_{dim} est le nombre de dimensions de la quantité de caractéristiques. Même si une constante est ajoutée ou soustraite de la valeur de pixel en raison d'un changement d'éclairage, la quantité de caractéristique est calculée par le gradient en raison de la différence de valeur de luminosité, elle n'a donc aucun effet. De plus, lorsque la valeur de pixel est multipliée ou divisée par une constante en raison de l'influence du changement d'éclairage, le changement de contraste peut être annulé par la normalisation vectorielle.

2.2.2. Speeded-Up Robust Features (SURF) Descriptor :

Semblable à SIFT, les fonctionnalités robustes accélérées (SURF) consistent en deux processus, la détection de points clés et la description de caractéristiques locales. Le processus de détection de points clés est décrit à la section 1.5.3. Ici, nous expliquons la description des caractéristiques locales, qui est le post-traitement de SURF.

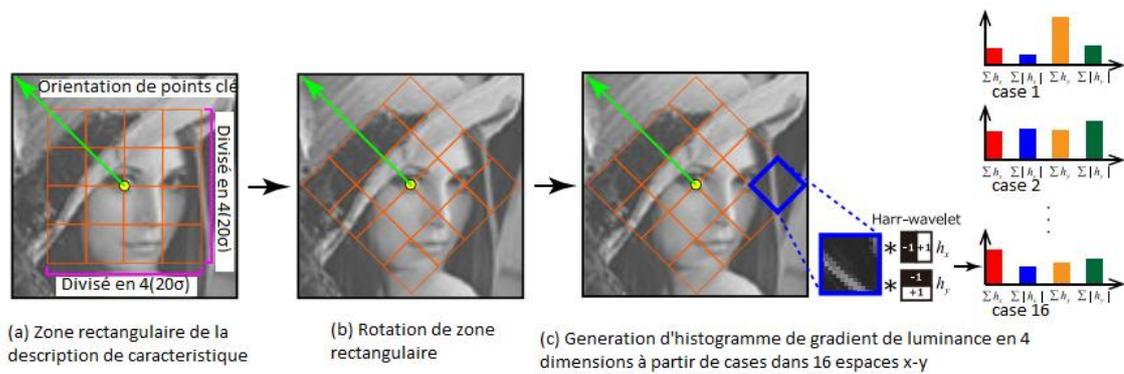


Figure 2.2. Description des caractéristiques SURF.

Pour la description de la caractéristique SURF, le gradient de luminosité est obtenu en utilisant Haar-wavelet comme dans le calcul de l'orientation. Comme le montre la figure 2.2 (a), la zone rectangulaire de $20\hat{\sigma} \times 20\hat{\sigma}$ centrée sur le point clé est divisée en une grille de 4×4 , et des cases sur l'espace x-y sont générées. La zone rectangulaire est tournée dans la direction d'orientation et le gradient de luminosité est calculé à l'aide d'une Haar-wavelet de $2\hat{\sigma} \times 2\hat{\sigma}$ pour chaque case dans l'espace x-y. Puisque SURF génère une image intégrée lorsqu'un point clé est détecté, le gradient de luminosité peut être obtenu à grande vitesse en utilisant Haar-wavelet. En supposant que les gradients de luminosité dans les directions x et y sont h_x et h_y , respectivement, un histogramme de gradient de luminosité à quatre dimensions $[\sum h_x, \sum h_y, \sum |h_x|, \sum |h_y|]$ est généré pour chaque case (Figure. 2.2(c)).

2.2.3. PCA-SIFT :

Les caractéristiques SIFT peuvent décrire des caractéristiques très précises, mais le problème est que la consommation de mémoire est importante car des caractéristiques de grande dimension à 128 dimensions sont requises pour chaque

point clé. PCA-SIFT [13] comprime la dimension des caractéristiques en utilisant l'analyse en composantes principales (PCA : principal component analysis).

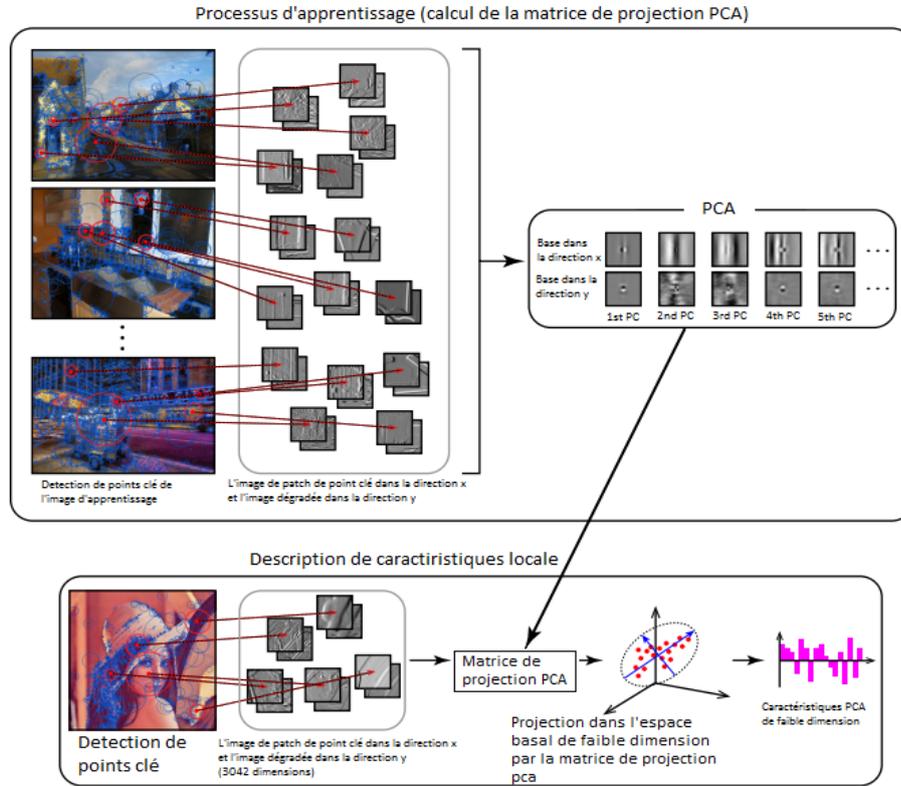


Figure 2.3. Description des caractéristiques de PCA-SIFT.

Tout d'abord, une image de la plage d'échelle des points clés détectés par SIFT est générée et redimensionnée à 41×41 pixels. Comme le montre la figure. 2.3, les gradients dans les directions x et y sont calculés à partir de l'image de patch redimensionnée et un vecteur dimensionnel de $39 \times 39 \times 2 = 3\,042$ est généré. Dans le calcul du dégradé, 1 pixel dans la zone de bord de l'image n'est pas utilisé, de sorte que chaque image dégradée est de 39×39 pixels. Ensuite, PCA est appliqué au vecteur de gradient de 3 042 dimensions pour générer la matrice de projection PCA P_G . La matrice de projection PCA $P_G \in R^{3042 \times N_p}$ est une matrice dans laquelle les images de patches de gradient x et y des points clés détectés à partir d'un grand nombre d'images d'apprentissage sont vectorisées, et les vecteurs propres correspondant aux valeurs propres N_p supérieures de la covariance matrice sont disposés. La référence [13] adopte $N_p = 36$. Le traitement jusqu'à ce point apprend à trouver la matrice de projection PCA P_G . Lors du calcul réel des caractéristiques locales, le point clé est le vecteur $d_{low} \in R^{N_p}$, qui est compressé dimensionnellement en multipliant le vecteur $g_{x-y} \in R^{3042}$ de l'image de patch de gradient x, y obtenue à partir des points clés par la matrice de projection PCA P_G . Il est utilisé comme grandeur de caractéristique locale.

$$d_{low} = P_G^T \cdot g_{x-y} \quad (2.2)$$

2.2.4. Gradient Location and Orientation Histogram (GLOH) :

Gradient Location and Orientation Histogram (GLOH) [35] est une extension des caractéristiques SIFT et il est conçu pour décrire des fonctionnalités plus robustes. Pour les caractéristiques telles que SIFT et SURF, l'espace x - y dans la zone autour du point clé est divisé en $4 \times 4 = 16$ cases en forme de grille pour décrire les caractéristiques. En revanche, GLOH transforme les cases dans l'espace x - y qui décrivent les caractéristiques en coordonnées log-polaires. Comme le montre la figure. 2.4 (a), l'histogramme de la direction du gradient est calculé à partir de 17 cases divisées en 3 dans la direction radiale et 8 dans la direction angulaire. En calculant l'histogramme de direction du gradient dans 16 directions dans chaque case, un vecteur de direction du gradient de $17 \times 16 = 272$ dimensions est calculé (Figure. 2.4 (c)). Ensuite, le vecteur de direction de gradient à 272 dimensions est compressé selon la même procédure que le PCA-SIFT décrit à la section 2.2.3, et enfin une caractéristique de direction de gradient à 128 dimensions est générée (Figure 2.4 (d)).

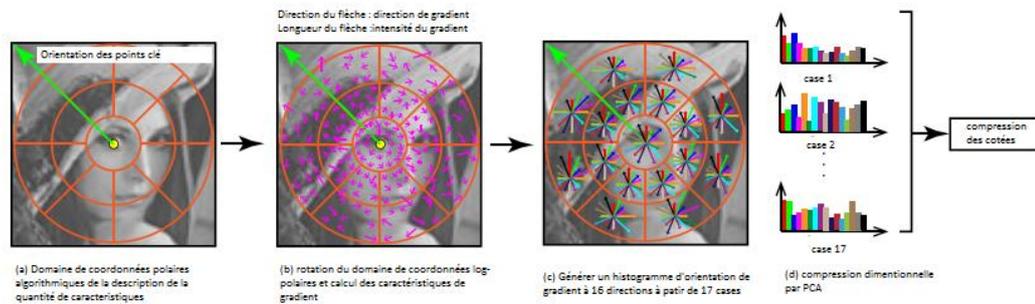


Figure 2.4. Description des fonctionnalités par GLOH

2.3. Description des caractéristiques locales par vecteur binaire :

Cette section décrit une méthode d'expression des caractéristiques locales par des vecteurs binaires. Bien que la précision des caractéristiques binaires soit inférieure à celle des caractéristiques réelles, la description des caractéristiques à grande vitesse et le calcul de distance sont possibles. Dans le calcul de distance, comme indique l'équation (2.11), il peut être calculé à grande vitesse par opération logique par XOR et comptage de bits.

2.3.1. Binary Robust Independent Elementary Features (BRIF):

Les caractéristiques élémentaires indépendantes robustes binaires (BRIF) [16] sont une méthode simple pour décrire les caractéristiques binaires à partir du signe de la différence de luminosité d'une paire de pixels sélectionnée au hasard à partir de l'image du patch autour du point clé. En supposant que la $i^{\text{ème}}$ paire de pixels sélectionnée au hasard dans l'image du patch est p_{u_i} et p_{v_i} et que la luminosité de

chaque pixel est $I(p_{u_i})$ et $I(p_{v_i})$, la caractéristique binaire d_i du $i^{ème}$ bit est donnée par ce qui suit peut être obtenue comme suit.

$$d_i = \begin{cases} 1 & \text{si } I(p_{u_i}) - I(p_{v_i}) > 0 \\ 0 (-1) & \text{sinon} \end{cases} \quad (2.3)$$

Les paires de pixels sélectionnées à partir de l'image sont préparées à l'avance sous forme d'ensembles N_{dim} , et il s'agit du nombre de dimensions de la quantité de caractéristiques. Il existe plusieurs façons de sélectionner les paires de pixels p_{u_i} et p_{v_i} , mais dans la référence [16], les pixels sont sélectionnés au hasard sur la base d'une distribution gaussienne pondérée au centre de l'image, comme le montre la figure 2.5 (a).

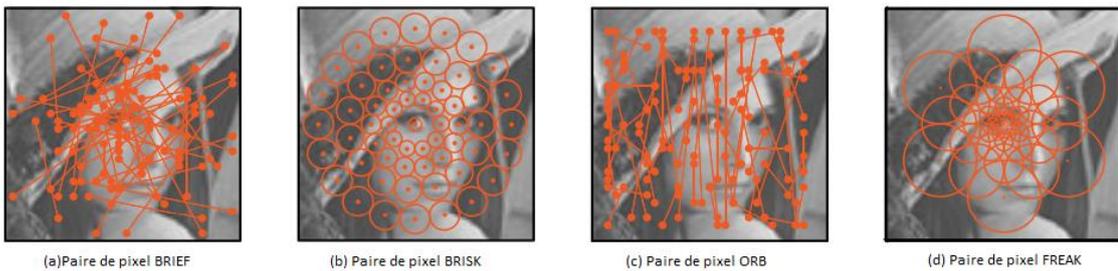


Figure 2.5. Modèle de paire de pixels pour les caractéristiques binaires.

En BRIEF, l'image est lissée au préalable avec un filtre Gaussien afin de réduire l'effet sur le bruit. De cette manière, la quantité de caractéristique peut être décrite uniquement en calculant la différence de luminance de la paire de pixels dans l'image pour le nombre de bits (nombre de dimensions), de sorte qu'un traitement à grande vitesse est possible.

2.3.2. Binary Robust Invariant Scalable Keypoints (BRISK) :

Binary Robust Invariant scalable Keypoints (BRISK) [17] utilise 60 valeurs de luminance également espacées sur quatre cercles concentriques placés dans l'image autour des points clés (Figure 2.5 (b)). BRIEF décrit dans la section 2.3.1 nécessite l'accès aux valeurs de luminance des pixels à $N_{dim} \times 2$ (par exemple, $N_{dim} = 256$) de la fonction, mais BRISK n'a besoin d'accéder qu'aux valeurs de luminance à 60 points. Chaque position d'échantillonnage est lissée par un filtre gaussien avec une variance proportionnelle à la distance au centre de l'image (cercle centré sur chaque position d'échantillonnage sur la figure. 2.5 (b)). BRISK propose une méthode d'estimation d'orientation unique. L'orientation est estimée à l'aide d'un ensemble de paires de pixels \mathcal{L}_{pair} (paire longue distance) où la distance de position d'échantillonnage est δ_{min} ou plus. La raison de l'utilisation de l'ensemble de paires à longue portée pour estimer l'orientation est de capturer la direction du gradient de luminance global dans l'image. Tout d'abord, dans la paire longue distance $p_i, p_j \in \mathcal{L}_{pair} (i, j = \{1, 2, \dots, 60\}, i \neq j)$, le gradient de luminance est calculé par l'équation suivante.

$$g_p(p_i, p_j) = p_i - p_j \frac{L(p_j; \sigma_j) - L(p_i; \sigma_i)}{\|p_j - p_i\|^2} \quad (2.4)$$

Ici, $L(p_i; \sigma_i)$ et $L(p_j; \sigma_j)$ sont les valeurs de luminance après lissage par un filtre gaussien d'échelle σ_i et σ_j . Comme le montre la Figure. 2.6 (a), g_p est la direction du gradient reliant les paires longue distance à une ligne droite, et l'amplitude du gradient est donnée par la différence de luminance des paires longue distance. Enfin, l'orientation globale $\hat{\theta}$ de l'image est estimée par l'équation suivante en utilisant le gradient obtenu par la paire longue distance.

$$\begin{aligned} [g_{p_x}, g_{p_y}] &= \frac{1}{|\mathcal{L}_{pair}|} \sum_{p_i, p_j \in \mathcal{L}_{pair}} g_p(p_i, p_j) \\ \hat{\theta} &= \tan^{-1} \left(\frac{g_{p_y}}{g_{p_x}} \right) \end{aligned} \quad (2.5)$$

Ainsi, l'orientation BRISK est définie comme l'angle du vecteur de gradient moyen de l'ensemble de paires à longue distance.

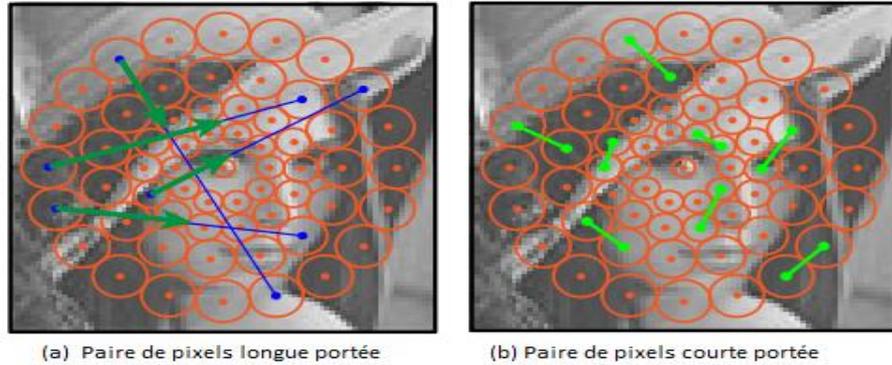


Figure 2.6 : Paires BRISK à longue et courte portée.

Ensuite, nous décrivons les caractéristiques binaires à l'aide d'un ensemble de paires de pixels S_{pair} (paire à courte distance) dont la distance de position d'échantillonnage est δ_{max} ou moins (Figure. 2.6 (b)). La raison d'utiliser l'ensemble de paires à courte portée pour la description des caractéristiques est de capturer les caractéristiques de l'image locale dans l'image. La caractéristique binaire d_i dans la $i^{\text{ème}}$ dimension peut être calculée par l'équation suivante.

$$d_i = \begin{cases} 1 & \text{si } L(p_j; \sigma_j) - L(p_i; \sigma_i) > 0 \\ 0(-1) & \text{sinon} \end{cases} ; \forall p_i, p_j \in S_{pair} \quad (2.6)$$

Puisque BIRSK utilise 512 paires à courte portée lors de la description des caractéristiques, une caractéristique binaire à 512 dimensions est finalement générée ($N_{dim} = 512$).

2.3.3. Oriented FAST and Rotated BRIEF (ORB) Descriptor :

Oriented FAST and Rotated BRIEF (ORB) [18] est également une méthode de description de caractéristiques binaires basée sur la différence de luminance de paires de pixels telles que BRIEF et BRISK. ORB propose non seulement une description des caractéristiques, mais également une méthode de détection de points clés combinant une pyramide d'image et un détecteur de coin FAST et une méthode d'estimation d'orientation originale. Les méthodes de détection des points clés et d'estimation de l'orientation dans l'ORB sont décrites dans la section 1.5.3. Ici, nous décrivons comment décrire les caractéristiques d'ORB.

Le procédé de description de quantité de caractéristiques par ORB est identique à BRIEF et BRISK, et le procédé de sélection de la paire de pixels dans l'image est conçu. Dans ORB, les paires de pixels qui améliorent statistiquement les performances de la description des caractéristiques pour toutes les paires de pixels sont étudiées et sélectionnées. Tout d'abord, un grand nombre d'images de points clés sont utilisées comme images d'apprentissage à l'avance, et 205 590 types de candidats de paires de pixels pouvant être inclus dans l'image de patch sont répertoriés. La différence de luminosité de toutes les paires de pixels candidats est calculée à partir de toutes les images d'apprentissage, et la paire de pixels est adoptée de telle sorte que la variance entre les images de patch soit grande et que la corrélation entre les paires de pixels soit faible. La variance E entre les images de patch est calculée par l'équation suivante.

$$E = \frac{1}{N_{img}^2} \sum_{m=1}^{N_{img}} \sum_{n=1}^{N_{img}} dist_H(d^{(m)}, d^{(n)}) \quad (2.7)$$

Ici, N_{img} est le nombre d'images de patch d'entraînement, et $d^{(m)}$ et $d^{(n)}$ sont des vecteurs de caractéristiques extraits de chaque image de patch d'entraînement. $dist_H(.)$ est une fonction qui calcule la distance de Hamming. La grande variance des caractéristiques entre les images de patch maximise la variance entre les classes, on peut donc dire que c'est une paire de pixels qui peut décrire des caractéristiques uniques dans diverses images. Si seules des paires de pixels avec une grande variance inter-classe sont utilisées, il est possible de sélectionner plusieurs fois des paires de pixels à des positions similaires dans l'image. Par conséquent, en ajoutant des paires avec une faible corrélation entre les paires de pixels aux conditions de sélection, les paires de pixels qui peuvent décrire de bonnes caractéristiques sont sélectionnées. La corrélation entre la $i^{ème}$ paire de pixels et la $j^{ème}$ paire de pixels $C_{i,j}$ est calculée par l'équation suivante.

$$C_{i,j} = \left| \frac{2}{N_{img}} \sum_{m=1}^{N_{img}} (d_i^{(m)} \oplus d_j^{(m)}) - 1 \right| \quad (2.8)$$

Le processus de sélection effective d'une paire de pixels est exécuté par la procédure suivante utilisant l'algorithme Greedy :

1. Pour toutes les paires de pixels candidats, triez par ordre décroissant de variance E entre les classes.
2. Pour toutes les paires de pixels candidats, la paire de pixels avec la plus grande variance inter-classe est adoptée.
3. Si la corrélation C est T_c ou moins dans toutes les paires de pixels adoptées. En se concentrant sur la paire de pixels candidate avec la deuxième plus grande variance après la paire de pixels adoptée, la corrélation C entre la caractéristique binaire de cette paire de pixels et la caractéristique binaire de la paire de pixels adoptée est calculée.
4. Le troisième processus est répété et lorsque les paires de pixels sont adoptées en 256 paires, le processus se termine.

Le seuil de la corrélation $C \in [0, 1]$ est initialement fixé à une valeur faible telle que $T_c = 0,2$. On recherche toutes les paires de pixels candidats, et si 256 paires de pixels ne sont pas adoptées, on augmente la valeur du seuil T_c et on répète à nouveau le troisième processus pour sélectionner la paire de pixels. La figure 2.5 (c) montre les paires de pixels sélectionnées par l'ORB. On peut voir que la paire de pixels sélectionnée est biaisée dans la direction verticale. En effet, l'image pour l'apprentissage est extraite par le détecteur de coin FAST et l'image de patch est corrigée dans la direction d'orientation, de sorte que la différence de luminosité de la paire de pixels dans la direction verticale est facilement adoptée comme quantité de caractéristique.

2.3.4. Fast Retina Keypoint (FREAK) :

Fast Retina Keypoint (FREAK) [19] sélectionne des paires de pixels en fonction de la structure rétinienne de l'organisme. Étant donné que la densité des cellules pyramidales dans la rétine du globe oculaire augmente à mesure qu'elle se rapproche du centre, FREAK utilise une position d'échantillonnage où la densité augmente à mesure qu'elle se rapproche du centre de l'image. La figure 2.5 (d) montre la position d'échantillonnage de FREAK. Le cercle centré à chaque position d'échantillonnage représente la plage d'échelle du filtre gaussien pour la réduction du bruit. L'échelle du filtre gaussien augmente de manière exponentielle à mesure que la distance depuis le centre de l'image augmente. Il y a 43 positions d'échantillonnage dans FREAK, ce qui est plus petit que celui de BRISK, mais il existe des milliers de combinaisons de toutes les paires de pixels. Par conséquent, dans FREAK ainsi que dans ORB, l'algorithme Greedy est utilisé pour sélectionner 512 paires de pixels qui augmentent la quantité d'informations dans les caractéristiques binaires. Parmi les 512 types de paires de pixels sélectionnés par FREAK, la paire de pixels extérieure est souvent sélectionnée dans la première moitié, et la paire de pixels près du centre de l'image du patch est souvent sélectionnée dans la seconde moitié.

Cette tendance est similaire au système visuel des organismes vivants. Dans le système visuel des êtres vivants, nous effectuons des observations approximatives dans le champ visuel périphérique et des observations plus détaillées dans le champ visuel central. Sur la base de la structure de ce système visuel, lors du calcul de la distance des caractéristiques binaires de 512 bits dans la recherche de points de correspondance, il est divisé en 4 parties tous les 128 bits pour construire une structure en cascade à 4 étages. Dans la structure en cascade, la distance est calculée dans l'ordre à partir des 128 bits supérieurs, et si la valeur de distance dépasse la valeur de seuil prédéfinie, le calcul est terminé et un rejet précoce est effectué. Lorsque la recherche de point correspondante est réellement effectuée à l'aide des fonctions FREAK, 90 % ou plus des points candidats correspondants sont rejetés uniquement dans les 128 premiers bits, de sorte qu'un traitement plus rapide est possible même dans la recherche de point correspondante.

2.4. Rechercher les points correspondants :

Dans la recherche de points de correspondance, les N_{dim} vecteurs dimensionnels $d, d' \in R^{N_{dim}}$ des caractéristiques locales calculées pour les points clés détectés entre les deux images sont comparés comme dans les équations (2.9) et (2.10). La distance euclidienne $dist_E$ et l'angle inter-vecteur $dist_\theta$ sont utilisés pour la comparaison entre les caractéristiques.

$$dist_E(d, d') = \sqrt{(d - d')^T (d - d')} \quad 2.9$$

$$dist_\theta = \cos^{-1} \left(\frac{d^T d'}{\|d\|_2 \|d'\|_2} \right) \quad 2.10$$

Si la caractéristique locale entre les deux images est un vecteur binaire $d, d' \in \{0; 1\}^{N_{dim}}$ composé de N_{dim} bits, la distance de Hamming $dist_H$ indiquée dans l'équation (2.11) est utilisée pour le calcul de la distance.

$$dist_H(d, d') = \text{bitcnt}(d \oplus d') \quad (2.11)$$

Ici, \oplus est un opérateur qui calcule XOR pour chaque élément du vecteur, et $\text{bitcnt}(\cdot)$ est une fonction qui compte les bits dans lesquels se trouve 1 du vecteur binaire. Lorsque les caractéristiques locales sont des vecteurs binaires, la distance entre les caractéristiques ne peut être calculée que par une opération logique et un simple comptage de bits, de sorte que la vitesse peut être augmentée même dans la recherche de points correspondante. La manière la plus simple de rechercher un point correspondant est une recherche par force brute de toutes les paires de caractéristiques entre deux images. Calculer la distance entre les caractéristiques d'un point clé détecté à partir de la première image et les caractéristiques de tous les points clés détectés à partir de la deuxième image, et utiliser la paire de caractéristiques avec la distance la plus proche. Trouver le 2ème voisin le plus proche $\{d, d''\}$, qui est la paire de caractéristiques avec la deuxième distance la plus proche d'un certain premier voisin le plus proche $\{d, d'\}$. Lorsque le rapport (ratio) des valeurs de distance du 1er plus proche voisin et du 2ème plus proche voisin est inférieur ou égal à la valeur de

seuil T_{ratio} , la paire de points clés du 1er plus proche voisin $\{d, d'\}$ est adoptée comme point correspondant.

$$\frac{\text{dist}(d, d')}{\text{dist}(d, d'')} \leq T_{\text{ratio}} \quad (2.12)$$

Le seuil T_{ratio} est déterminé par l'application. Par exemple, lorsque T_{ratio} est petit, le nombre de points de correspondance erronés peut être réduit, ce qui est efficace pour la reconnaissance d'objet basée sur le vote. Si le T_{ratio} est grand, le nombre de points de correspondance erronés augmentera, mais comme les points de correspondance globaux augmenteront, il est efficace pour le traitement d'alignement entre les images utilisant RANSAC [25] etc., qui peut tolérer certaines valeurs aberrantes.

2.5. Conclusion :

Dans ce chapitre, les descripteurs de caractéristiques locales proposés jusqu'à présent sont décrits.

La description des caractéristiques locales, une correspondance de points clés haute performance peut être réalisée par la méthode de description des caractéristiques basée sur l'histogramme de direction du gradient de la région autour des points clés [1, 36]. De plus, de nombreuses méthodes pour décrire des caractéristiques de faible dimension et robustes ont été proposées en compressant des caractéristiques de grande dimension et en simplifiant les histogrammes [9, 13, 23]. De plus, en conservant la quantité de caractéristiques sous la forme d'un vecteur binaire au lieu de la conserver sous la forme d'un vecteur de nombre réel, une économie de mémoire et une correspondance de points clés à grande vitesse peuvent être réalisées, de sorte qu'elles ont fait l'objet de recherches énergiques. [16, 17, 18, 20, 19, 24].

CHAPITRE III

| Expérimentation et résultats

Dans ce chapitre et après l'aperçu théorique dans les chapitres précédents, nous présentons le côté pratique de notre travail. Notre but est la comparaison d'un ensemble d'algorithmes de détection et de description des points clés. Ce chapitre décrit l'environnement de développement, le langage de programmation, ainsi que le processus de tests et d'évaluations de notre approche. Dans la suite du chapitre nous présentons les résultats enregistrés ainsi que l'évaluation de ces derniers.



3.1. Expérimentations :

3.1.1. Environnement de développement :

a. Environnement matériel :

Le système informatique utilisé est un ordinateur LenouvoG50 avec les caractéristiques suivantes :

Processeur	Intel (R) Pentium (R) CPU N3540 (2.16GHz)
Mémoire (RAM)	4.0 Go

Tableau 3.1. Description de matériel

b. Environnement logiciel :

Système d'exploitation	Windows 7 Professionnel (64 bits)
Environnement de développement	Jupyter notebook + spyder
Langage de programmation	Python 3.6

Tableau 3.2. Description de logiciel

Dans le processus de développement on a utilisé les modules python cité au dessus :

- **OpenCv[37] :**

Open Source Computer Vision Library est une bibliothèque ouverte sous licence BSD qui comprend plusieurs centaines d'algorithmes de vision par ordinateur. OpenCV a une structure modulaire, ce qui signifie que le paquet comprend plusieurs bibliothèques partagées ou statiques. Les modules suivants sont disponibles :

- **Corefunctionality (Fonctionnalité de base) :** Un module compact définissant les structures de données de base, y compris la matrice multidimensionnelle dense Mat et les fonctions de base utilisées par tous les autres modules.
- **Traitement d'image (imgproc) :** Un module de traitement d'image qui comprend le filtrage d'image linéaire et non linéaire, les transformations d'image géométriques (redimensionnement, distorsion affine et perspective, remapping générique basé sur table), la conversion d'espace de couleur, histogrammes, et ainsi de suite.
- **2D Features Framework (features2d) :** les détecteurs de caractéristiques principales, descripteurs et mise en correspondance de descripteur est sous ce module et c'est le module utilisable dans notre travail.

- **Pandas [38]:**

Le nom "Pandas" a une référence à la fois "Panel Data", et "Python Data Analysis" et a été créé par WesMcKinney en 2008. Elle est une bibliothèque libre qui permet des fonctions pour analyser, nettoyer, explorer et manipuler les données en proposant en particulier des structures de données et des opérations de manipulation de tableaux numériques et des séries.

- **Numpy [39] :**

Numpy est une bibliothèque libre et open source pour le langage python, elle est destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

- **Matplotlib [40]:**

Matplotlib est une bibliothèque complète pour créer des visualisations statiques, animées et interactives en Python. Matplotlib rend les choses faciles et les choses difficiles possibles.

3.1.2. Les bases d'images utilisées:

Nous avons adopté deux bases pour nos tests de comparaison des algorithmes de mise en correspondance local. Ces bases de données sont un ensemble d'images de cellules histologiques.

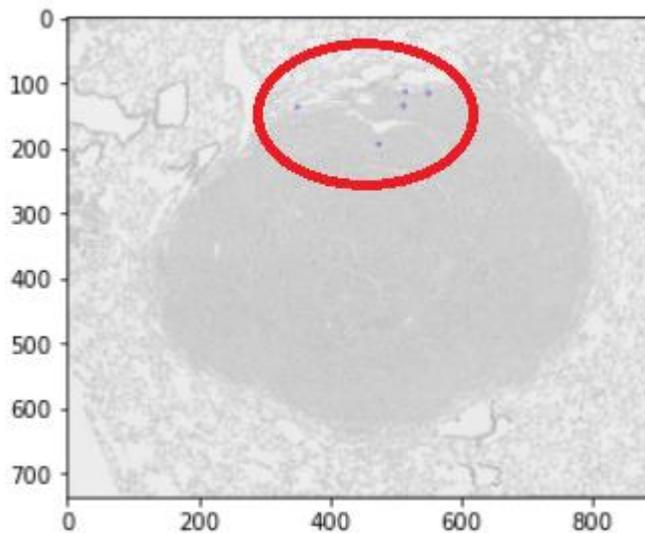
L'ensemble de données se compose de tranches de tissu de microscopie histologique 2D, colorées avec différentes taches, et de points de repère indiquant les points clés de chaque tranche. Les principaux défis pour ces images sont les suivants : très grande taille d'image, différences d'apparence et absence d'objets d'apparence distinctifs. Des repères placés manuellement pour l'évaluation de la qualité du matching, sont enregistrés dans des fichiers « CSV ».

Un échantillon de tissu de cinq coupes de tache avec des points de repère en surbrillance est illustré ci-dessous (Figure 3.1).

Des tranches de tissu consécutives ont été colorées avec plusieurs colorants différents, voir, voir Colorants et marqueurs moléculaires.

Les teintures utilisées sont les suivantes :

- Protéine des cellules de Clara 10 (Cc10)
- Protéine C prosurfactante (proSPC)
- Hématoxyline et éosine (H&E)
- Antigène KI-67 (Ki67)
- Molécule d'adhésion des cellules endothéliales plaquettaires (PECAM-1, également connue sous le nom de CD31)
- Récepteur 2 du facteur de croissance épidermique humain (c-erbB-2/HER-2-neu)
- Récepteur des œstrogènes (ER)
- Récepteur de la progestérone (PR)



(a) Le fichier : 29-041-lzd2-w35-Cc10-5-les1.csv

Unnamed: 0	X	Y
0	1	85 140
1	2	139 165
2	3	177 99
3	4	228 194
4	5	190 241

(b) Le fichier : 29-041-lzd2-w35-Cc10-5-les1.csv

Figure 3.1. Un échantillon de tissu de cinq coupes de tache avec des points de repère en surbrillance.

3.1.3. Processus de travail :

Pour l'évaluation des algorithmes de mise en correspondance locale, la méthodologie est illustrée à la figure 3.2.

La première partie du processus est la détection des points d'intérêts, et la deuxième partie du processus c'est la recherche des correspondances.

a. Détection et description des points d'intérêts :

Comme était mentionné dans les premiers chapitres il existe différentes méthodes pour détecter et décrire les points clés d'une image. Dans notre travail on a décidé de choisir 6 algorithmes :

- **SIFT** et **SURF** qui ont l'avantage d'être détecteurs de points clés à l'aide d'espace d'échelle, et parce que ce sont des descripteurs de caractéristiques locales par vecteurs réels.
- **BRISK** qui est un algorithme de détection de points clés avec invariance d'échelle et invariance de rotation, et il est aussi un descripteur de caractéristiques locales par vecteurs binaires.
- **ORB** qui est une fusion du détecteur de points clés FAST et du descripteur BRIEF.
- **BRIEF** et **FREAK** sont que des descripteurs de caractéristique binaire.
- **FAST**, pour détecter les points clés utilisés par BRIEF et FREAK.

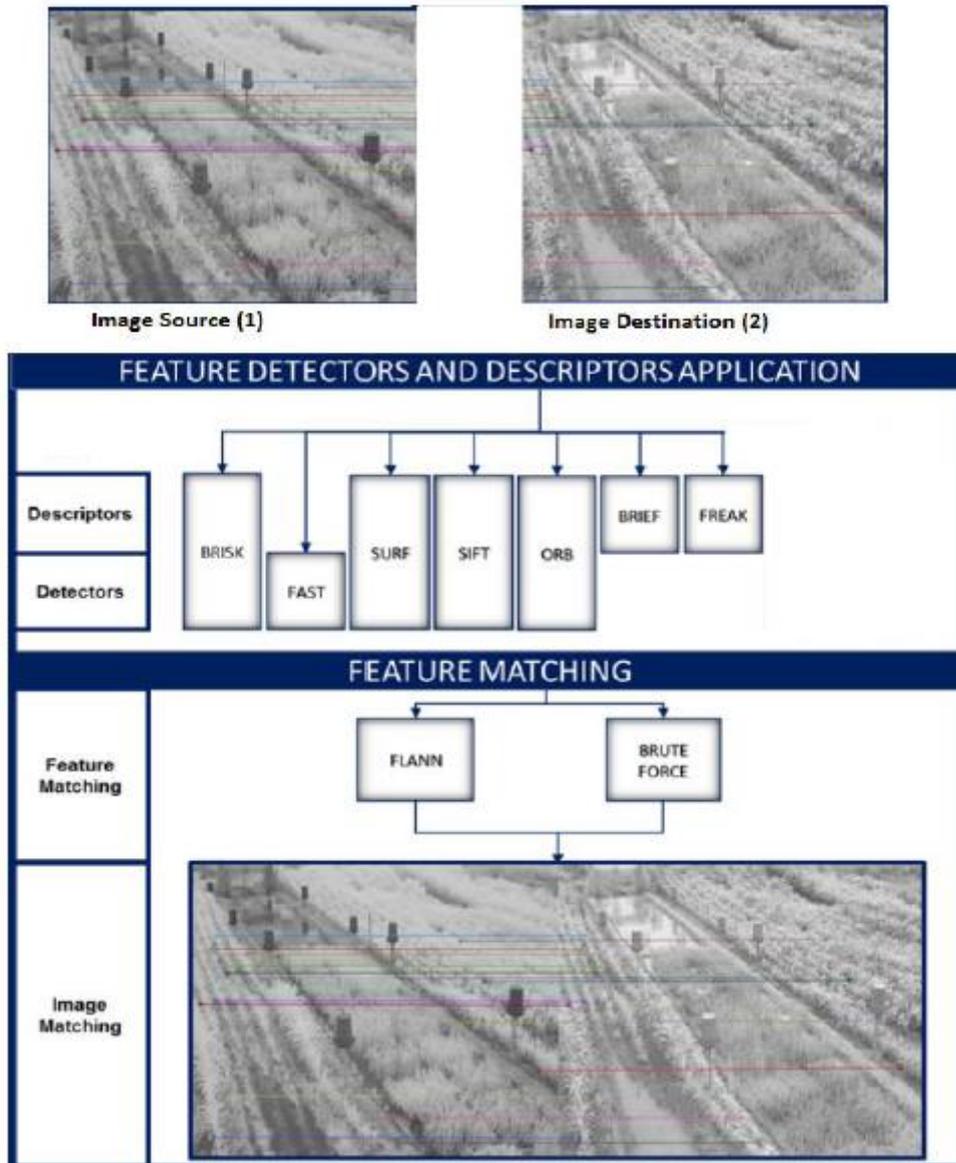


Figure 3.2: La méthodologie de test

b. La recherche des correspondances :

La bibliothèque OpenCV fournit deux méthodes qui calcul la correspondance entre deux images. C'est le cas de FLANN et Brute Force.

- **Brute Force** : Il prend le descripteur d'une caractéristique dans le premier ensemble et est mis en correspondance avec toutes les autres caractéristiques du deuxième ensemble en utilisant un calcul de distance et le plus proche est retourné.

La section 2.4 présente les différentes formule pour le calcul des correspondances ; Nous trouverons une implémentation adéquate dans OpenCv (voir tableau 3.3)

Descripteur	Distance Correspondante	Fonction OpenCv
SURF	Norme L1	NORM_L1
SIFT		
BRISK		
ORB	Norme Hamming	NORM_HAMMING2
BRISK	Norme Hamming	NORM_HAMMING
FAST + BRIEF		
FAST + FREAK		

Tableau 3.3 Distance BF matching pour descripteurs des caractéristiques

- **FLANN** : Fast Library for Approximate Nearest Neighbors signifie Bibliothèque rapide pour les voisins les plus proches approximatifs. Il contient une collection d'algorithmes optimisés pour la recherche rapide du voisin le plus proche dans de grands ensembles de données et pour les caractéristiques de grande dimension.

c. Les tests :

Nous avons effectué deux tests sur l'ensemble de nos bases de données.

- **Le premier test** : concerne la puissance de détection des différents algorithmes dans ce type d'images, pour cela nous avons passé par l'ensemble des étapes du processus à savoir la détection le calcul des descripteurs et la recherche des correspondances.
- **Le deuxième test** : concerne la qualité des descripteurs. Pour cela nous avons utilisé les fichiers CSV, qui contiennent des points repères dans chaque image (les coordonnées sont pré calculé). Dans ce deuxième test, nous n'avons pas besoin de l'étape de détection dans les algorithmes, nous utilisons seulement les deux autres étapes à savoir le calcul des descripteurs et la recherche des correspondances.

d. Evaluation des résultats :

Comme les deux tests diffèrent les méthodes d'évaluation aussi diffèrent.

Dans le premier teste nous savons comparer la distance moyenne des correspondances et le temps d'exécution entre FLANN et brute force matcher (BF) pour chaque descripteur. Puisque le nombre de correspondance est trop élevée qui a rendu difficile l'évaluation de la qualité des correspondances, on a décidé de ne prendre que les résultats des vingt correspondances les plus significatifs obtenus entre chaque paire d'images avec les différentes algorithmes. La précision est donnée par la moyenne des correspondances. Cette mesure donne une bonne idée de la qualité des correspondances obtenues avec chaque technique.

Concernant le deuxième test, on a utilisé plus de distance moyenne et le temps d'exécution les métriques suivants :

$$Recall = \frac{nbr\ correct\ matches}{nbr\ correspondances} \quad (3.1)$$

$$1 - Precision = \frac{nbr\ faux\ matches}{nbr\ correct\ matches + nbr\ faux\ matches} \quad (3.2)$$

- *nbr correct matches* est le nombre de vraies correspondances,
- *nbr faux matches* est le nombre de fausses correspondances
- *nbr correspondances* est le nombre total de correspondance trouvé.

3.2. Résultats et discussion :

3.2.1. Présentation des images :

Pour chaque base d'images les figures 3.i(1), figures 3.i(2), ($i = 3, 4, 5, 6$) présentent les images aux niveaux de gris, avec les point clés visualisé a l'aide des points bleu. Les figures 3.i(3), ($i = 3, 4, 5, 6$) présentes les correspondances des images calculé par la méthode Brute Force (BF) et les figures 3.i(4), ($i = 3, 4, 5, 6$) présentes les correspondances des images calculé par la méthode FLANN.

3.2.2. Teste N° 1 :

a. Détection et correspondances des points clé :

Cette section présentes les images avec les points clé pour les algorithmes BRISK, ORB, SIFT et SURF aussi les 20 premier correspondances des images utilisant BF et FLANN.

- **BRISK** : On remarque que l'algorithme détecte plusieurs points clés. BF matcher a détecté des correspondances, au contraire de Flann qui n'a détecté aucune.

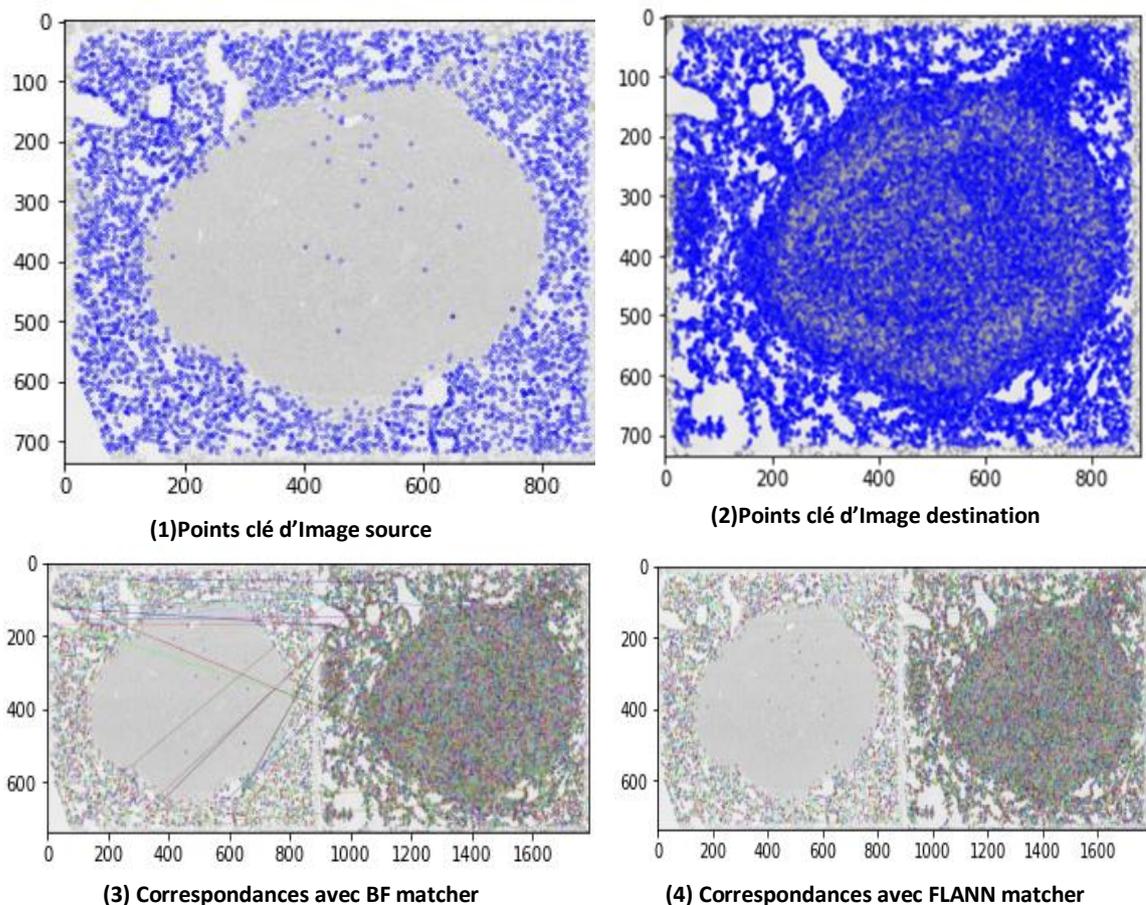


Figure 3.3. Test 1 du descripteur BRISK

- **ORB** : Le nombre des points clé détecté est moins par rapport au celle de BRISK, SIFT, SURF. La méthode FLANN détecte 0 correspondances.

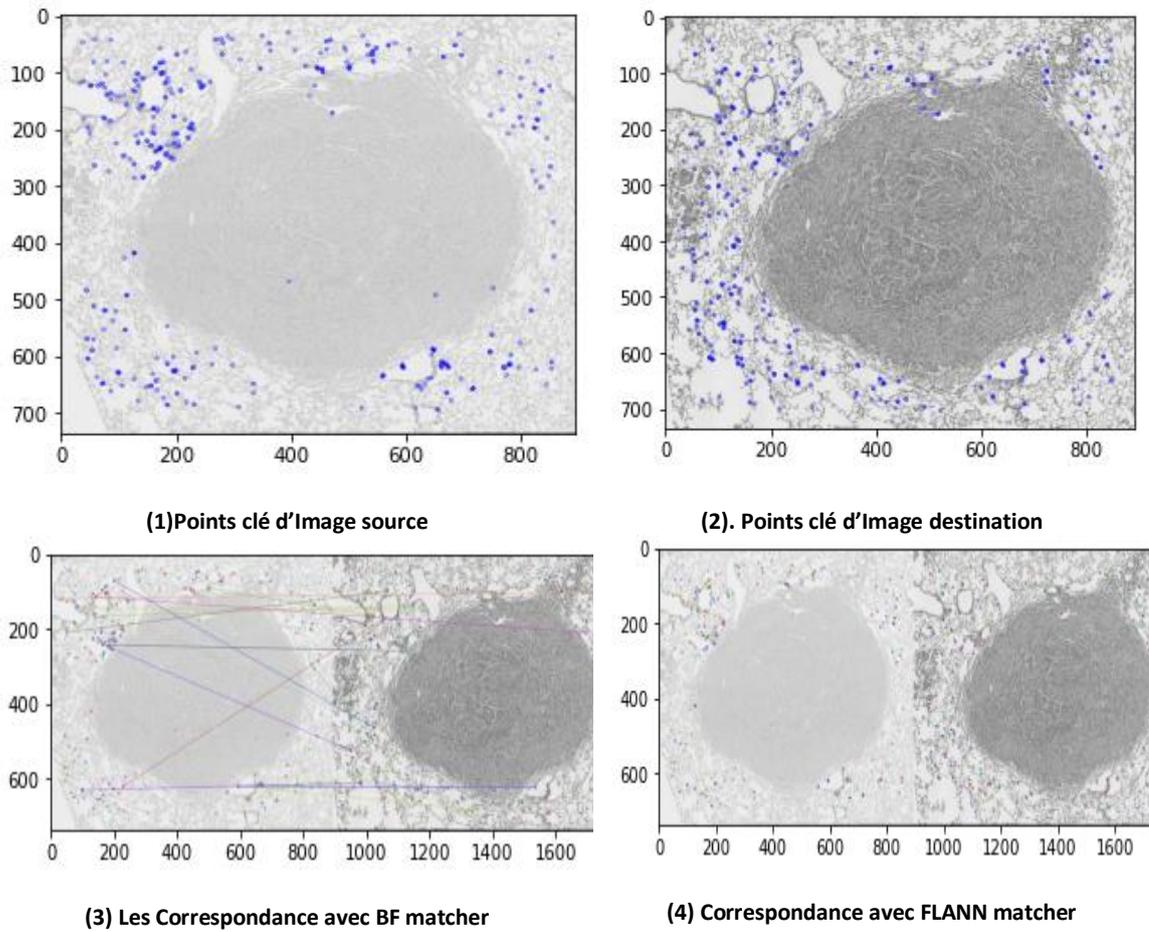
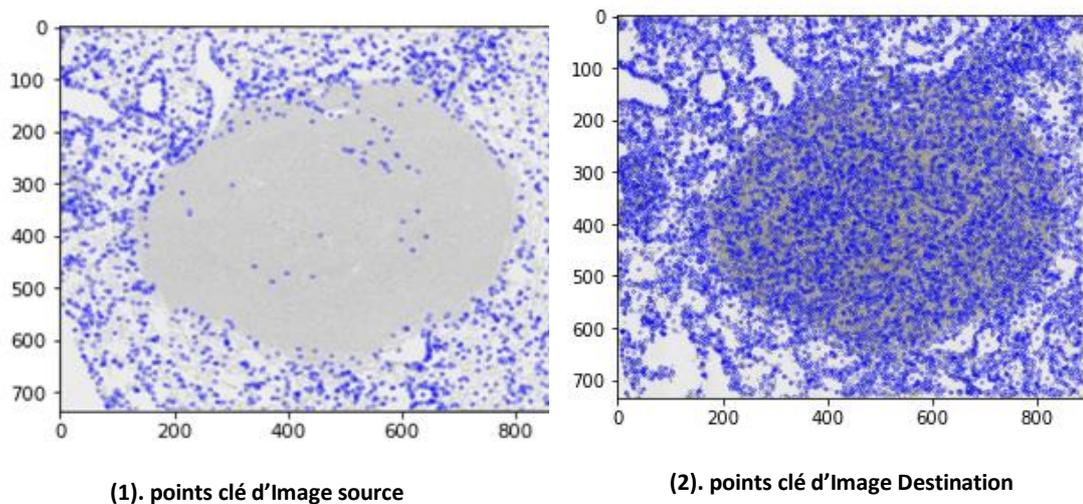


Figure 3.4. Test 1 du descripteur ORB

- **SIFT** : Le nombre de point clés est important, mais le nombre des correspondances avec FLANN est moins que avec BF-Matcher.



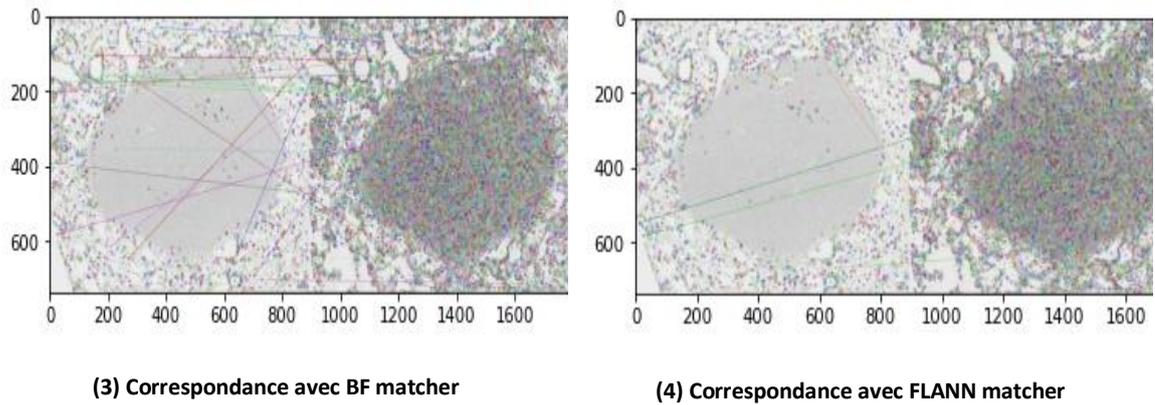


Figure 3.5. Test1 du descripteur SIFT

▪ SURF :

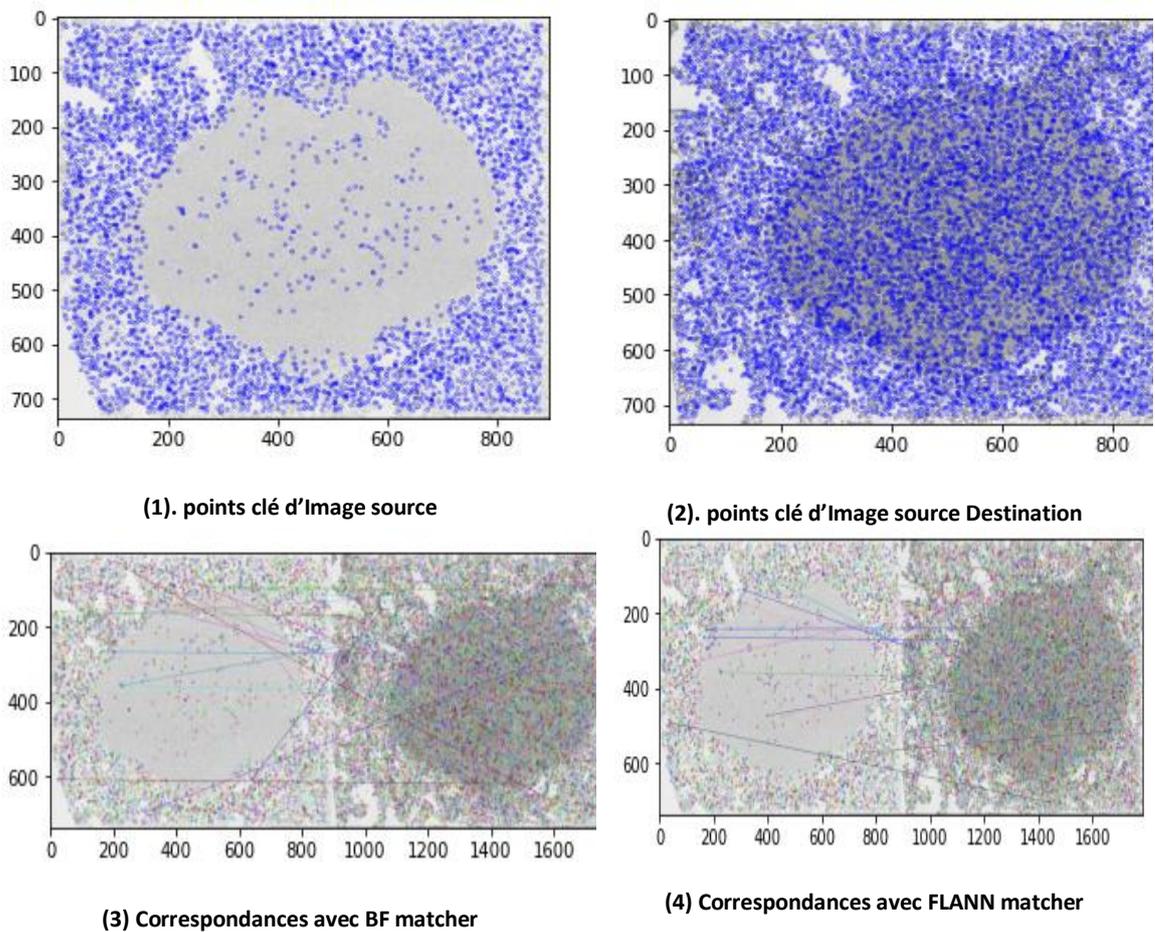


Figure 3.6. Test1 du descripteur SURF

b. Evaluation des résultats :

L'évaluation des teste sert à comparer le temps d'exécution et les distances moyenne des correspondances calculé en changeant la méthode du matching. Pour mieux comprendre voir les graphes suivants (figure 3.7 et figure 3.8 et figure 3.9).

Nous remarquons sur les premiers graphes, qui représentent les résultats enregistré pour chaque test (chaque recherche des correspondances entre 2 images),

que le temps d'exécution pour BF-Matcher dépend de la complexité des deux images, et cela pour tous les descripteurs. Par contre pour FLANN cette remarque n'est valide que pour le descripteur SURF, par contre pour les autres descripteurs les changements ont non significatifs.

Dans l'évaluation des distances moyenne, La méthode FLANN enregistre une grande variation entre les résultats enregistré pour chaque test (chaque recherche des correspondances entre 2 images), au contraire de BF-Matcher qu'elle enregistre une stabilité dans la distances moyennes entres ces même tests. Cela revient à la nature de travail des deux méthodes, puisque BF-Matcher cherche toujours la meilleure solution.

▪ **FLANN :**

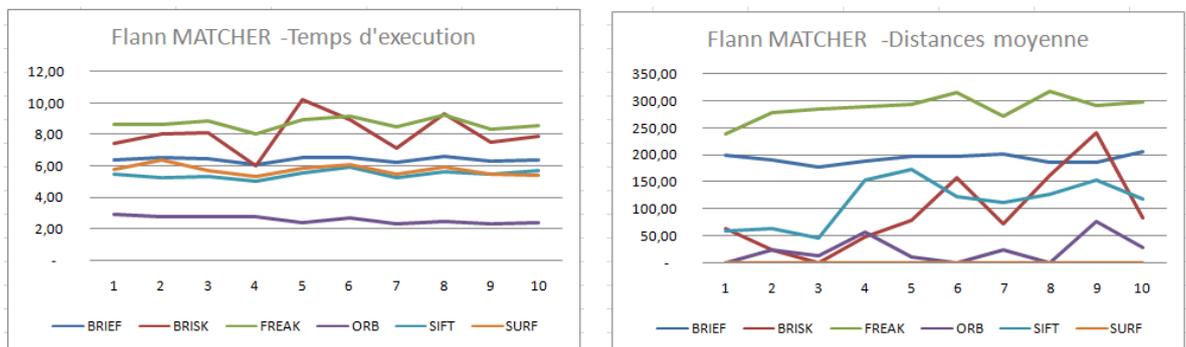


Figure 3.7. Resultat de Test 1 utilisant FLANN

▪ **Brute Force (BF) :**

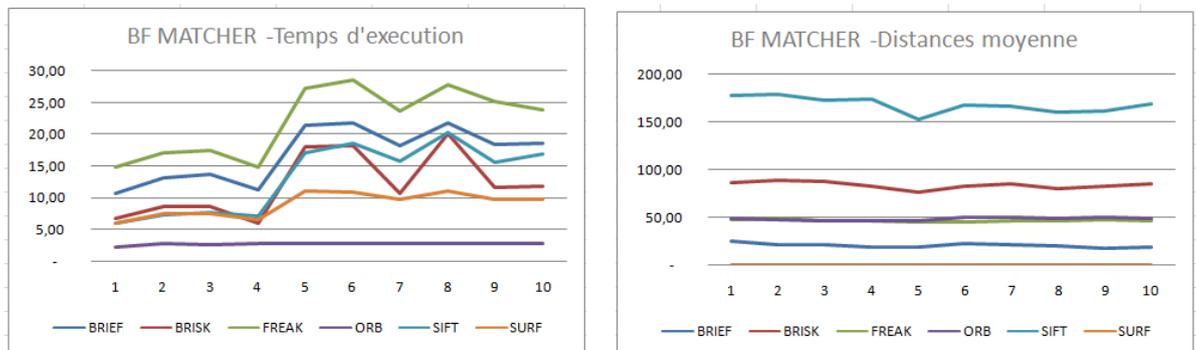


Figure 3.8. Resultat de Test 1 avec Brute Force

▪ **Comparaison BF et FLANN :**

Les résultats enregistrés montrent bien la supériorité de FLANN par rapport à BF-Matcher dans la vitesse de recherche des correspondances, et ce si pour tous les descripteurs. Par contre, la qualité des correspondances trouvées est moins bonne, vue les distances moyennes trouvées. Seuls les descripteurs SFIT et ORB font l'exception, et FLANN dans les meilleurs résultats.

	Distance moyenne					
	BRIEF	BRISK	FREAK	ORB	SIFT	SURF
BF MATCHER	20,75	84,31	46,76	48,38	168,12	0,14
Flann MATCHER	192,96	93,65	287,43	23,45	113,71	0,15

Tableau 3.4. Comparaison de distance moyenne entre BF et FLANN

	Temps d'exécution					
	BRIEF	BRISK	FREAK	ORB	SIFT	SURF
BF MATCHER	16,95	12,06	22,03	2,79	13,24	8,95
Flann MATCHER	6,39	8,05	8,66	2,57	2,48	5,75

Tableau 3.5. Comparaison de temps d'exécution entre BF et FLANN

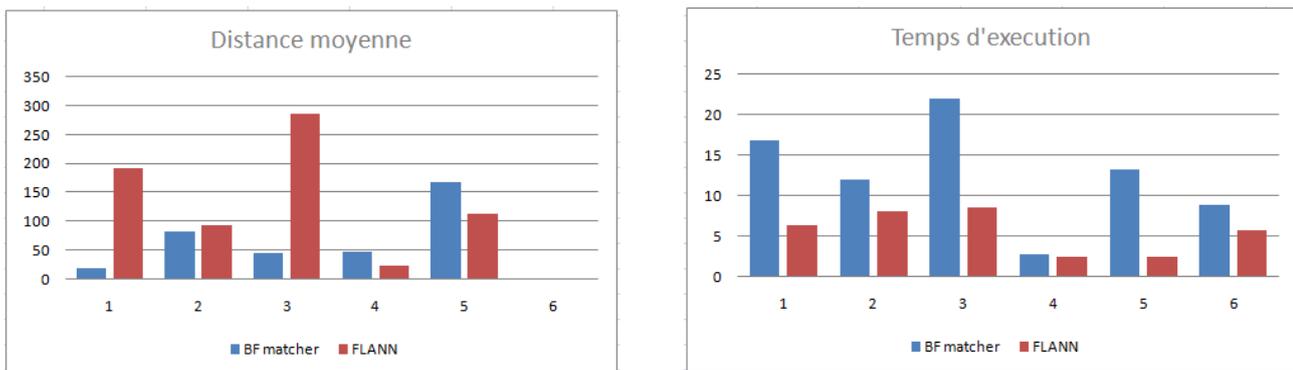


Figure 3.9. Comparaison de temps d'exécution et distance moyenne entre BF et FLANN

3.2.3. Teste N° 2 :

a. Détection des points clé :

Pour ce test nous avons utilisé les fichiers (.CSV) qui contiennent les coordonnées des points clés pour chaque image. Le nombre de ces points clés est 78. La figure 3.9 représente les mêmes points clé utilisé pour chaque descripteur.

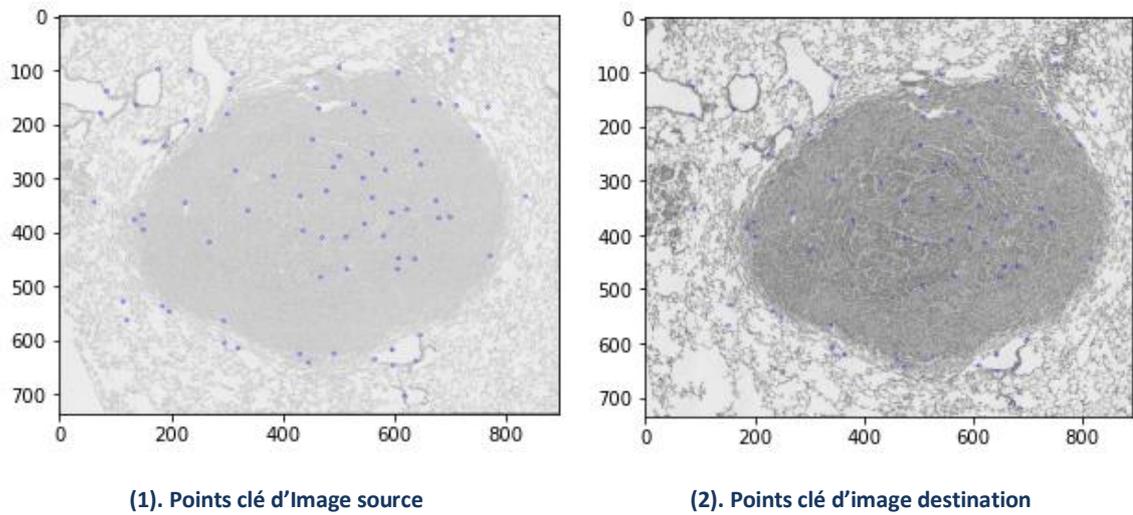


Figure 3.9. Points clé pour teste 2

b. Correspondance des points clé :

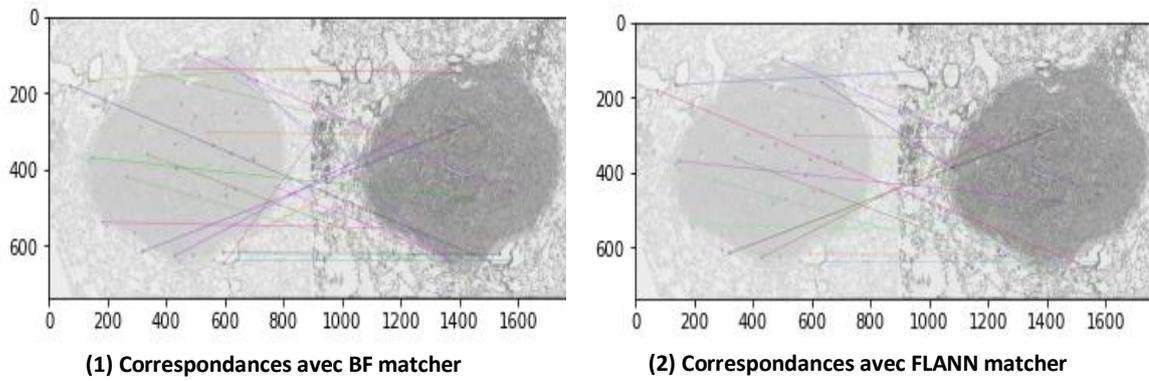


Figure 3.10. Test2 du descripteur BRISK

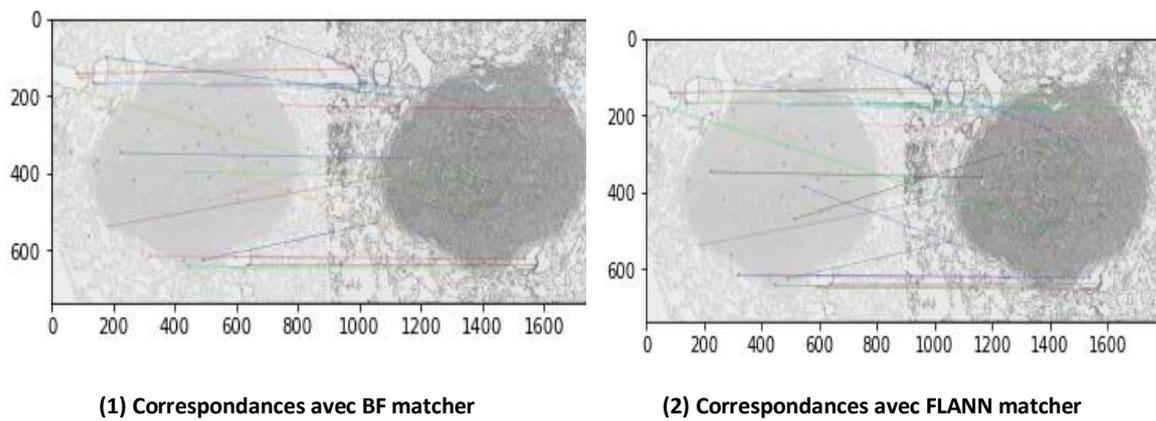


Figure 3.11. Test2 du descripteur ORB

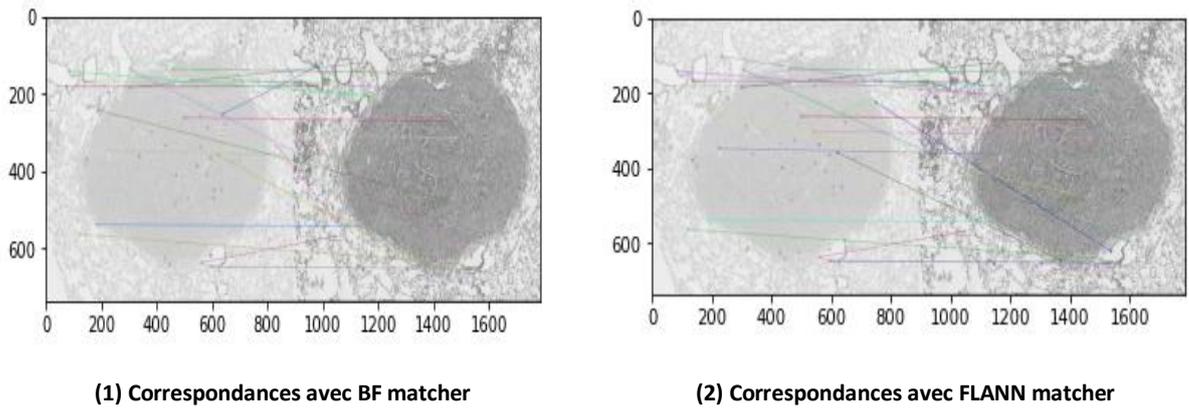


Figure 3.12. Test2 du descripteur SIFT

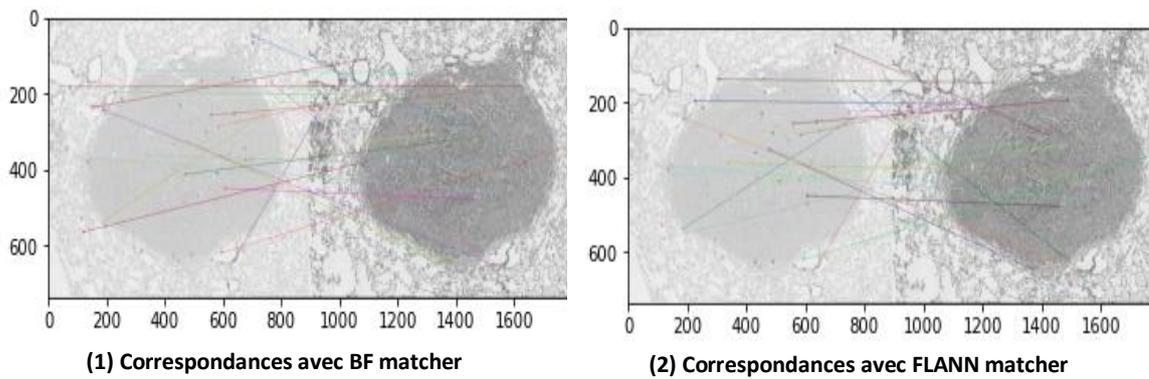


Figure 3.13. Test2 du descripteur SURF

a. Evaluation des résultats :

Après le calcul des correspondances on a remarqué que les valeurs de tableau 3.6 sont identiques pour Brute Force et FLANN, la seule différence mineure réside dans le temps d'exécution. Voir graphe 3.14.

Descripteur	Nombre des matchings trouvées	Nombre des vrais matching (Positive)	Nombre des faux matching Negative	recall	1-Precision	distance Moyenne
BRIEF	46	9	37	0,12	0,81	65,89
BRISK	45	5	40	0,07	0,88	159,34
FREAK	43	3	39	0,04	0,92	123,32
ORB	45	6	39	0,08	0,86	63,13
SIFT	48	10	38	0,13	0,79	334,37
SURF	46	2	44	0,03	0,95	0,17

Tableau 3.6 : résultats du Test N°2 , BF et Flann

Les résultats dans les tableaux sont explicables par le nombre réduits des points clés utilisé pour la recherche des correspondances. Toutefois, le nombre maximum des correspondances (Matching) trouvée par les descripteurs est de 48, il ne représente que 61%

des correspondances existantes (78 correspondances exacte). Ainsi que le nombre de fausses correspondances est plus grand que le nombre de vraies correspondances. C'est seulement SIFT et BRIEF qui ont donnés les meilleurs résultats, un recall de 13% et 12% respectivement.

Il faut noter que SIFT à donner des résultats sur toutes les images de tests de la DataSet1, au contraire des autres descripteurs qui dans quelque images il non pas pu faire des correspondances, exemple BRIEF ne donne pas de résultat avec le test 3, ainsi que le descripteur BRISK avec le test 7.

	BRIEF	BRISK	FREAK	ORB	SIFT	SURF
FLANN	2.21	3.46	2.41	2.23	2.30	2.18
BF	2.18	3.38	3.39	2.20	2.29	2.18

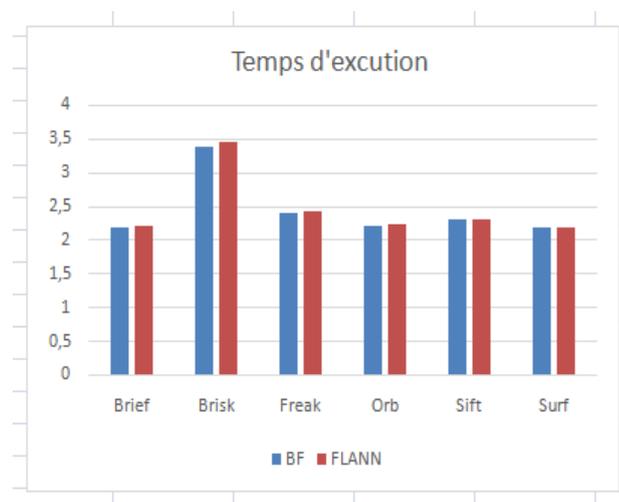


Figure 3.14 : résultats du Test N°2 ; différence entre BF et Flann

Les résultats obtenus montrent clairement que l'utilisation des descripteurs classiques sur les images histologiques ne donne pas de bons résultats. Ceci est à cause de à la nature même de ces images, il y a une grande différence d'apparence entre les images, et l'absence d'objets d'apparence distinctifs rend difficile de trouver des caractéristiques dans les images. Nous pensons que l'utilisation d'une pile de prétraitement sur les images est nécessaire pour aider à renforcer la qualité de recherche des correspondances. Dans cette pile nous pouvons suggérer une normalisation des intensités entre les différentes images.

3.3. Conclusion :

Dans ce chapitre, nous avons exposé les résultats obtenues par l'ensemble des descripteurs de la littérature que nous avons comparée. Nous avons validé une série de tests sur des images histologiques. Cette série d'expérimentations que nous Dans ce chapitre, nous avons exposé les résultats obtenues par l'ensemble des descripteurs de la littérature que nous avons comparée. Nous avons validé une série de tests sur des images histologiques. Cette série d'expérimentations que nous avons menée, nous a permis de constater que les descripteurs testés ne sont pas appropriée à l'utilisation sur les images histologiques.

Conclusion Général et perspectives :

Dans ce manuscrit, nous nous sommes intéressés à l'évaluation des algorithmes de détection et de description des points d'intérêt. Ce type d'algorithmes est couramment utilisé dans des applications telles que la reconstruction 3D, l'imagerie médicales, le suivi d'objets ou encore la reconnaissance de gestes.

Nous avons tout d'abord réalisé une étude bibliographique recensant un certain nombre de méthodes existantes, afin de lister leurs avantages et inconvénients. Nous avons pu établir différents constats. Concernant les détecteurs de points d'intérêt, le détecteur de Harris reste le plus utilisé. Le FAST-hessien et le ORB ont quant à eux l'avantage de posséder une analyse multi-échelle, permettant notamment d'observer différents niveaux de détails. La description du voisinage du point d'intérêt s'appuie sur des histogrammes de gradients orientés (SIFT) ou des ondelettes de Haar (SURF). Ils permettent tous deux d'analyser les différents gradients constituant le voisinage du point. L'étape de mise en correspondance repose le plus souvent sur la minimisation de la distance inter-descripteur. Cette dernière permet de déterminer les couples de points avec la meilleure ressemblance.

Cette étude nous a permis de tester la qualité de six descripteurs, et de deux méthodes de mise en correspondance. Le choix d'utiliser les images d'histologie pour faire les tests, est encouragé par les défis que relèvent ce type d'images. Une grande taille, et l'absence d'objets d'apparence distinctifs dans les images.

Les nombreux tests et résultats détaillés au chapitre 3 mettent en avant les performances des différents algorithmes. Nous avons constaté, que les résultats obtenus montrent clairement que l'utilisation des descripteurs classiques sur les images d'histologie ne donne pas de bons résultats.

Les travaux réalisés au cours de cette thèse nous ouvrent un large champ de perspectives. D'une part, proposer l'utilisation d'une pile de prétraitements, avant la phase de calcul des descripteurs. Cette pile devra réduire les différences dans les intensités entre les images. Enfin, le développement d'un nouveau descripteur spécialement pour ce type d'imagerie.

Références :

1	D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol.60, no.2, pp.91–110, 2004.
2	H. Moravec, "Towards Automatic Visual Obstacle Avoidance", International Joint Conference on Artificial Intelligence, p.584, 1977.
3	H. Moravec, "Rover visual obstacle avoidance", International Joint Conference on Artificial Intelligence, pp.785–790, 1981.
4] P. R. Beaudet, "Rotationally invariant image operators", International Conference on Pattern Recognition, pp.579–583, 1978.
5	C. Harris, and M. Stephens, "A combined corner and edge detector", Alvey Vision Conference, pp.147–151, 1988
6	S. M. Smith, and J. M. Brady, "Susan & mdash; a new approach to low level image processing", International Journal of Computer Vision, vol.23, no.1, pp.45–78, 1997.
7	T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales", Journal of Applied Statistics, pp.224–270, 1994.
8	T. Lindeberg, "Feature Detection with Automatic Scale Selection", International Journal of Computer Vision, vol.30, pp.79–116, 1998.
9] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded-Up Robust Features", Computer Vision and Image Understanding, vol.110, no.3, pp.346–359, 2008.
10	K. Mikolajczyk, and C. Schmid, "Indexing Based on Scale Invariant Interest Points", International Conference on Computer Vision, pp.525–531, 2001.
11	G. Koutaki, and K. Uchimura, "Scale-space Processing Using Polynomial Representations", Conference on Computer Vision and Pattern Recognition, pp.2744–2751, 2014.
12	K. Mikolajczyk, and C. Schmid, "Scale & Affine Invariant Interest Point Detectors", International Journal of Computer Vision, vol.60, no.1, pp.63-86, 2004.
13	Y. Ke, and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors", Conference on Computer Vision and Pattern Recognition, vol.2, pp.II–506, 2004.
14	S. Wu, and M. S. Lew, "RIFF: Retina-inspired Invariant Fast Feature Descriptor", ACM international conference on Multimedia, pp.1129–1132, 2014.
15] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo", Pattern Analysis and Machine Intelligence, vol.32, no.5, pp.815–830, 2010.
16	C. Michael, L. Vincent, S. Christoph, and F. Pascal, "BRIEF: binary robust independent elementary features", European Conference on Computer Vision, pp.778–792, 2010.
17	S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints", International Conference on Computer Vision, pp.2548–2555, 2011.

18	E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF", International Conference on Computer Vision, pp.2564-2571, 2011.
19	A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast Retina Keypoint", Conference on Computer Vision and Pattern Recognition, pp.510–517, 2012.
20	T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, "Boosting binary keypoint descriptors", Conference on Computer Vision and Pattern Recognition, pp.2874–2881, 2013.
21	K. Min, L. Yang, J. Wright, L. Wu, X. S. Hua, and Y. Ma, "Compact projection: Simple and efficient near neighbor search with practical memory requirements", Conference on Computer Vision and Pattern Recognition, pp.3477–3484, 2010.
22	D. Achlioptas, "Database-friendly Random Projections: Johnson-Lindenstrauss with Binary Coins", Journal of Computer and System Sciences, vol.66, no.4, pp.671–687, 2003.
23	M. M. Bronstein, C. Strecha, A. M. Bronstein, and P. Fua, "LDAHash: Improved Matching with Smaller Descriptors", Pattern Analysis and Machine Intelligence, vol.34, pp.66-78, 2011.
24	M. Ambai, and Y. Yoshida, "Card: Compact and real-time descriptors", International Conference on Computer Vision, pp.97–104, 2011.
25	R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. M. Frahm, "USAC: a universal framework for random sample consensus", Pattern Analysis and Machine Intelligence, vol.35, no.8, pp.2022–2038, 2013
26	J. Shi, and C. Tomasi, "Good Features to Track", Conference on Computer Vision and Pattern Recognition, pp.593 - 600, 1994.
27	J. P. Gravel, "Corner Detection", Biological Cybernetic, vol.59, no.4, pp.139 - 153, 1988.
28	T. Tuytelaars, and K. Mikolajczyk, Local Invariant Feature Detectors: A Survey, Now Publishers Inc., 2008
29	Yasushi Kanazawa, Kenichi Kanaya, "Extraction de points caractéristiques d'image pour l'explication de la vision par ordinateur", Journal de l'Institut des ingénieurs en électronique, en information et en communication, vol.87, no.12, pp.1043–1048, 2004
30	E. Rosten, R. Porter, and T. Drummond, "FASTER and better: A machine learning approach to corner detection", Pattern Analysis and Machine Intelligence, vol.32, pp.105–119, 2010.
31	J. R. Quinlan, "Induction of decision trees", Machine Learning, vol.1, no.1, pp.81–106, 1986.
32	P. Perona, "Steerable-scalable kernels for edge detection and junction analysis", European Conference on Computer Vision, pp.3–18, 1992.
33] D. Shy, and P. Perona, "X-Y separable pyramid steerable scalable kernels", Conference on Computer Vision and Pattern Recognition, pp.237–244, 1994.
34	W. Freeman, and E. Adelson, "The design and use of steerable filters", Pattern Analysis and Machine Intelligence, vol.13, pp.891–906, 1991
35	K. Mikolajczyk, and C. Schmid, "A performance evaluation of local descriptors", Pattern Analysis and Machine Intelligence, vol.27, no.10, pp.1615–1630, 2005.

36	R. Arandjelovic, and A. Zisserman, "Three things everyone should know to improve object retrieval", ' Conference on Computer Vision and Pattern Recognition, pp.2911–2918, 2012.
37	Introduction. <i>openCv</i> . [En ligne] 04 07 2018. https://docs.opencv.org/3.4.2/d1/dfb/intro.html .
38	<i>Pandas</i> . [En ligne] [Citation : 07 03 2022.] https://pandas.pydata.org/
39	<i>Numpy</i> . [En ligne] [Citation : 14 07 2022.] https://numpy.org/ .
40	<i>Matplotlib: Visualization with Python</i> . [En ligne] [Citation : 14 07 2022.] https://matplotlib.org/

ملخص

تُستخدم طرق رؤية الكمبيوتر في الوقت الحاضر في العديد من التطبيقات، على سبيل المثال المراقبة بالفيديو أو التصوير الطبي أو إعادة البناء ثلاثي الأبعاد. بحيث تركز هذه الأخيرة على عمليات التعرف على الأشكال و النماذج. للقيام بذلك، يتم تحليل الصورة لاستخراج المعالم أو العناصر الأولية (المحيط أو وظائف شدة الإضاءة أو النماذج المورفولوجية). تعتمد الطرائق الحالية على استعمال النقاط الأساسية التي تمثل عدم استمرارية تدرجات الرمادي التي دورها تحدد الأركان أو الزوايا الموجودة في الصورة. يتم استعمال توصيف محلي من أجل القيام بمطابقة مجموعة من النقاط الأساسية من صورة لصور أخرى. تسمح هذه التقنية باستخراج معلومات حول المنطقة المجاورة لكل نقطة (قيم البيكسل و شدة الإضاءة و الميل). يهدف هذا العمل البحثي المتمثل في هذه الأطروحة إلى إعطاء نظرة حول مختلف الأدوات المستعملة للكشف عن النقاط الأساسية ووصفها ومطابقتها. أولا نقتراح تحليلا بيبليوغرافيا للطرق الموجودة حاليا , بحيث تمكننا هذه الأخيرة من استنتاج نهج التنفيذ بالإضافة إلى المزايا والعيوب الرئيسية. بعد ذلك نتحدث بالتفصيل حول الطريقة المقترحة من أجل تقييم فاعلية الخوارزميات المستعملة. يتم استخدام قاعدتي بيانات تصوير طبي للأنسجة في التجارب المقدمة. الخوارزميات المستعملة للكشف عن العناصر الأولية ووصفها هي (SIFT , SURF, BRISK, ORB, BRIEF , FREAK et FAST). أما بالنسبة لإيجاد التناظر بين النقاط الرئيسية فأساسه الطريقتين BF-Matcher و Flann .

الكلمات المفتاحية : مطابقة, توصيف, BF-Matcher, Flann, علم الانسجة

Abstract

Today computer vision methods are used in many applications such as video monitoring, medical imaging or 3D reconstruction. These different applications generally rely on pattern recognition or tracking processes. To do this, the image is analyzed in order to extract landmarks or primitives (contours, intensity functions or morphological models). The most common methods are based on the use of key points representing a discontinuity of gray levels characterizing a in an image. In order to match a set of points from one image to another, a local description is used. It makes it possible to extract information from the neighborhood of each point (values of pixels, light intensities, gradients).

The research work presented in this thesis aims to implement different tools for detecting, describing and matching key points. First, we propose a bibliographical analysis of existing methods that makes it possible to deduce the implementation parameters as well as the main advantages and disadvantages.

We then detail the process proposed for evaluating the performance of the algorithms used. Two histology medical imaging datasets are used for the tests. The detection and description of the primitives is based on the algorithms (SIFT, SURF, BRISK , ORB, BRIEF , FREAK and FAST). The matching of key points is based on the BF-Matcher and Flann methods.

Keywords : Matching, Descriptors, BF-Matcher, FLANN, Histology

Résumé

De nos jours les méthodes de vision par ordinateur sont utilisées dans de nombreuses applications telles que la vidéosurveillance, l'imagerie médicale ou la reconstruction 3D par exemple. C'est différentes applications s'appuient généralement sur des procédés de reconnaissance de formes ou de suivi. Pour ce faire, l'image est analysée afin d'en extraire des amers ou des primitives (contours, fonction d'intensité ou modèles morphologiques). Les méthodes les plus courantes s'appuient sur l'utilisation de points d'intérêts représentant une discontinuité des niveaux de gris caractérisant un coin dans une image. Afin de mettre en correspondance un ensemble de points d'image à une autre, une description locale est utilisée. Elle permet d'extraire l'information du voisinage de chaque point (valeur des pixels, des intensités lumineuses, des gradients). Les travaux de recherche présentée dans ce mémoire ont pour objectif de mettre en œuvre différents outils de détection, description et mise en correspondance de points d'intérêts. Dans un premier temps, nous proposons une analyse bibliographique de méthodes existantes. Cette dernière permet en effet déduire les paramètres de mise en œuvre ainsi que les principaux avantages, et inconvénients. Nous détaillons par la suite le processus proposé pour évaluer des performances des algorithmes utilisé. Deux bases d'images médicales sont utilisées pour les tests. La détection et le description des primitives repose sur les algorithmes (SIFT, SURF, BRISK, ORB, BRIEF, FREAK et FAST) La mise en correspondance des points d'intérêts se base quant a elle sur les méthodes BF-Matcher et FLANN.

Mots clé : Mise en correspondance, Descripteurs, BF-Matcher, FLANN, Histologie