République Algérienne Démocratique et Populaire

Ministaire de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique

Département d'Électronique

Projet de fin d'etude en vue de l'obtention du Diplome d'Ingénieur

d'État en Électronique

Étude comparative d'algorithmes de reconnaissance d'objets: SIFT,SURF,ORB et SFOP
Application aux modèles de voitures

Réalisé par :
BOUBEZARI Rayana
OUDNI Louiza

Proposé et Encadré par :
Pr C.Larbes
Pr A.Bouridane

Promotion 2013

ملخص

التعـــرف علـى الأشـــياء الموجـودة علـى الصـــور أحد أهم تطبيقـــــات الرؤيـــة الاصـــــطناعية.

هناك العديـــد من الطـرق و الأســـــاليب الـــتي تهـدف إلـى الكشـــف والتعـــرف علـى الأشـــياء علـى الصـــور .

مشـــروعنا عبـــارة عن مقارنـــة لعـدة خوارزميـات محليـــة ، الـــتي تعتمـــد أساســـا علـى وصـــائف و نقـــاط مثـــيرة للاهتمـــام لوصـــف الصـــورة .

إن هذه المقارنـــة تســـمح لأي مســـتخدم الاختيـــار بيـــن مختلـــف الأســـاليب القائمـــة ، وفقـــا لتطبيقـــــه .

درســـنا هذه الخوارزميـــات نظريـــا و طبقناهـــا علـى قاعـدة تحتـــوي علـى صـور نمـاذج لســـيارات .

لقـد أظهرنا من خلال الإحصـــــائيات الـــتي أجريـت علـى برنـــامج متـلاب Matlab ، أداء كل خوارزميـة علـى وضـــعيات مختلفـــة للصـــورة.

كلمـات مفتاحيـــــة : الكشـــف ، التعـــرف ،SIFT، ORB، SFOP، SURF، واصـف، نقطـة مثـيرة للاهتمـام، مقيـــاس ، اتجـــاه.

Résumé

La vision par ordinateur est devenue un domaine incontournable de l'intelligence artificielle. Parmi les applications les plus importantes de ce domaine, nous citons la reconnaissance d'objet.

Il existe différentes approches et plusieurs méthodes dont le but est la détection et la reconnaissance d'objet sur les images. L'une des approches ayant donnée de bons résultats récemment est l'approche locale. L'objectif de ce projet consiste à comparer les algorithmes suivants : SIFT, SURF, SFOP et ORB selon plusieurs critères. Ces algorithmes sont basés sur l'approche locale. Ils extraient des points d'intérêts et calculent des descripteurs afin de caractériser une image. Cette comparaison permettrait à un éventuel utilisateur de choisir entre les différentes méthodes existantes, selon le contexte de son application.

Nous avons étudié ces algorithmes théoriquement et nous les avons appliqués à une base de données contenant des images de modèles de voitures.

Nous avons montré grâce à des statistiques effectuées sur Matlab, les performances de chacun des algorithmes pour différentes variations de l'image.

Mots clés : reconnaissance, détection, SIFT, SURF, SFOP, ORB, point d'intérêt, descripteur, orientation, échelle, orientation.

Abstract

Computer vision has become an essential field of artificial intelligence. Among the most important applications of this area, we include object recognition.

There are different approaches and several methods aimed at the detection and object recognition.One of the recent approaches that gives good results is the local approach.

The objective of this project is to compare the folowed algorithms : SIFT, SURF, SFOP and ORB, according to many criteria. These algorithms are local, and extract interest points and calculate descriptors to characterize an image.This comparison would allow a potential user to choose between the various existing methods, depending on the context of its application.

We studied these algorithms theoretically and we applied them to a car model images database.

We have shown through statistics performed on Matlab, the performance of each algorithm with respect to different variations of the image.

Keywords : recognition, detection, SIFT, SURF, SFOP, ORB, interest point, descriptor, orientation, scale.

Remerciements

Nous tenons d'abord à exprimer notre gratitude à nos promoteurs, au Pr.BOURIDANE pour nous avoir initiées au domaine du traitement d'image, et au Pr.LARBES pour la disponibilité et l'écoute permanentes dont il a fait preuve tout au long de ce projet.

Nous tenons à remercier les membres du jury pour l'intérêt accordé à notre travail.

Enfin, nous remercions tous ceux qui auront contribué à ce travail et à notre cursus universitaire.

Dédicaces

Je dédie ce travail à :

Mes très chers parents, ce travail représente le fruit de vos investissements,

À ma sœur, pour le soutien et l'encouragement prodigués,

À mon frère, mes oncles et tantes,

À mes amis,

Et à tout le corps de l'école nationale polytechnique.

Rayana

Dédicaces

Je dédicace ce travail à :
Ma très chère famille,
Mes chers amis,
Aux membres de la famille de l'Ecole Polytechnique.

Louiza

Table des matières

Table des figures

Liste des tableaux

Abreviations

BRIEF	**B**inary **R**obust **I**ndependent **E**lementary **F**eatures
DoG	**D**ifference **o**f **G**aussians
FAST	**F**eatures from **A**ccelerated **S**egment **T**est
KLT	**K**anade **L**ucas **T**omasi
ORB	**O**riented FAST **R**otated **B**RIEF
SFOP	**S**cale invariant **Feature OP**erator
SIFT	**S**cale **I**nvarant **F**eature **T**ransform
SURF	**S**peeded **U**p **R**obust **F**eatures
SUSAN	**S**mallest **U**nivalue **S**egment **A**ssimilating **N**ucleus

Introduction générale

La vision par ordinateur est à la croisée des chemins entre les mathématiques, le traitement du signal et l'intelligence artificielle. C'est une science qui procure aux ordinateurs et aux machines, la faculté de capturer des images, d'en extraire les caractéristiques essentielles, de les interpréter et de les exploiter.

L'objectif de ce projet est la comparaison de plusieurs algorithmes de traitement d'image, conçus spécialement pour la détection et la reconnaissance d'objets sur des images, ou encore le suivi d'objets sur des vidéos. Ces algorithmes sont le SIFT, le SURF, le SFOP et l'ORB. Ils se basent sur la détection de points clés et le calcul des descripteurs pour caractériser l'image.

Le premier chapitre présente un état de l'art sur les algorithmes de reconnaissance d'objet. Il permet de positionner les méthodes abordées parmi les techniques existant dans la littérature.

Dans le deuxième chapitre, nous étudions l'aspect théorique des méthodes comparées. Nous exposons aussi les différents outils mathématiques utilisés.

Le dernier chapitre est consacré aux résultats pratiques obtenus. Nous avons appliqué les méthodes précédemment citées à une base de données. Les images de cette base représentent des voitures, le but étant de reconnaitre de manière automatique les modèles de celles-ci. Nous avons ainsi calculé leurs pourcentages de reconnaissance et également testé leur invariance par rapport à différentes variations de l'image. Nous avons alors évalué leurs pourcentages de reconnaissance en augmentant le bruit sur l'image, en diminuant la qualité, en variant l'échelle, la luminosité et la rotation, ou encore en occultant l'objet sur l'image.

Chapitre 1

Etat de l'art

1.1 Introduction

Au milieu des années cinquante est née l'intelligence artificielle, discipline qui a pour ambition de faire faire aux machines des raisonnements similaires à l'intelligence humaine. Tout comme l'homme, la machine doit être capable notamment de voir et d'interpréter les données acquises. Ce domaine de recherche porte le nom de " vision par ordinateur". L'un des axes les plus importants de ce domaine, est la reconnaissance d'objet. C'est d'ailleurs le thème auquel nous introduit ce chapitre.

1.2 Les défis majeurs de la reconnaissance d'objet

La difficulté majeure du problème de la reconnaissance d'objets repose sur les variations d'un objet ou des objets d'une même catégorie sur des images différentes. On peut grouper les sources de variations dans les catégories suivantes [1] [2] :

- **Les variations de point de vue**
  Les objets réels sont des entités physiques tridimensionnelles. Leur apparence change de façon significative en fonction du point de vue.

- **Les variations d'illumination**
  En fonction de changements de l'environnement et des conditions d'acquisition d'une image, un objet peut subir des variations d'illumination importantes.
  Les objets dans les images des scènes naturelles sont très sensibles aux variations

d'illumination. Celles-ci modifient considérablement les couleurs et les valeurs radiométriques des objets.
Une modification uniforme sur l'ensemble de l'image peut être annulée par une rectification de la luminosité et du contraste à des valeurs standards. Cependant, dans un cas réel les changements ne sont pas uniformes. Par exemple quand le soleil se déplace, quand des rideaux s'ouvrent, ou quand une lampe s'allume.

- **Les occultations**
  Dans une image les objets peuvent être partiellement masqués par d'autres objets. Les occultations de parties d'objets sont une grande source de variabilité de l'apparence des objets. Les apparences typiques de parties d'objets sont remplacées par l'objet occultant.

- **Les variations d'échelle**
  Les objets d'une même catégorie ou les images d'un même objet peuvent avoir des tailles différentes en fonction de la résolution de l'image.

- **Les déformations**
  La plupart des objets réels n'ont pas une forme rigide. Ceci génère de grands changements d'apparence entre deux états. Nous pouvons citer par exemple de petites apparitions/disparitions comme une antenne radio rétractable sur une voiture. Ou encore, des déformations articulées que subit un pantin articulé ou des déformations plastiques comme un visage qui fait une grimace.

- **Complexité du fond**
  En général, dans une image les objets ne sont pas isolés de leur environnement.La position des objets dans les images est souvent déterminée par un rectangle englobant, aussi appelé Region Of Interest (ROI). Ce rectangle contient l'intégralité de l'objet, et aussi des pixels qui proviennent du fond de scène. Quand ce fond de scène change, le contenu du rectangle englobant change aussi, ce qui est une nouvelle source de variabilité.

- **Variations intra-classe**
  Les objets d'une même classe peuvent avoir des apparences très variables.

1.3 Différentes approches

On peut distinguer trois types d'approches, permettant la reconnaissance d'objets [1].

1.3.1 Méthodes géométriques

Historiquement, ce sont les premières méthodes de reconnaissance d'objet. Elles représentent les objets par leurs contours. On peut distinguer parmi ces méthodes, celles à base d'alignement modèle-objet.
Les méthodes d'alignement géométrique disposent d'un modèle 3D de l'objet recherché. Les primitives (par exemple des droites) composant ce modèle 3D sont alignées sur les primitives détectées dans l'image. Ainsi, la qualité de l'alignement détermine si l'objet est présent ou non .[3]
Comme ces méthodes ne considèrent que les contours des objets, elles ignorent donc leur intérieur. Cela leur permet d'obtenir de la robustesse par rapport au changement d'illumination et de la texture des objets.
Cependant, certaines catégories d'objet, comme les arbres, ne peuvent être définies uniquement par leurs contours. De plus, elles ne sont pas robustes à une détection imprécise des contours, ce qui est souvent le cas dans des conditions d'acquisition d'images réelles.

1.3.2 Méthodes globales

Les méthodes de reconnaissance globale calculent une signature de l'image prise dans sa globalité. Cette signature peut être, par exemple, la distribution des couleurs dans l'image. Elle peut aussi être constituée de l'ensemble des pixels mis les uns à la suite des autres dans un vecteur colonne.
La comparaison avec les images de référence peut se faire avec une analyse en composante principale, ce qui est le cas des eigenfaces de Turk et Pentland [4] par exemple.
Ces méthodes ont l'avantage d'être simples, et robustes aux modifications d'illumination quand la base d'images d'apprentissage contient une grande quantité d'images aux conditions d'illumination différentes. Mais elles obtiennent de mauvais résultats en cas d'occultation et de changement d'arrière plan.

1.3.3 Méthodes locales

Contrairement aux méthodes globales, les méthodes locales ne considèrent plus les images comme un tout monolithique. Elles le considèrent plutôt comme une collection de régions locales, qui sont le plus souvent des parties d'images carrées ou rectangulaires.
Ces régions locales s'affranchissent des difficultés rencontrées sur des images entières, car elles peuvent être ramenées à une apparence standard. Les modifications d'illumination

sont localement uniformes, et la luminosité et le contraste peuvent donc être normalisés [5].
Et comme ces régions sont de petites tailles, elles sont le plus souvent présentes intégralement ou bien totalement occultées. Elles sont rarement partiellement occultées. Leur apparence ne varie donc pas à cause des occultations.
Le principe de ces méthodes est de représenter les images de référence par des régions locales et de stocker les descripteurs de ces régions locales (par exemple leurs niveaux de gris) dans une base de données.
Lors de la reconnaissance, les descripteurs des régions locales sélectionnées dans l'image sont comparés aux descripteurs des images de la base de données. Cela permet de sélectionner l'image de référence qui contient les descripteurs les plus proches de celle de l'image à reconnaitre.
Les méthodes locales sont rapides et ont une bonne gestion des variations citées plus haut. Elles ont aujourd'hui la préférence de la communauté pour la reconnaissance d'objets.

1.4 Définition d'un point clé

Les méthodes utilisées dans le contexte de ce projet, se basent sur des points clés. Un point clé [6] ou un point d'intérêt est un point d'une image qui diffère de ses voisins immédiats. Il diffère en terme de certaines propriétés spécifiques de l'image, telles que l'intensité, la couleur, l'orientation, la texture, la courbure, (figure 1.1). La détection de points d'intérêt est l'un des thèmes les plus importants du domaine de la vision par ordinateur. Ces points constituent des caractéristiques utiles dans les problèmes d'appariement d'images, de reconnaissance d'objets.

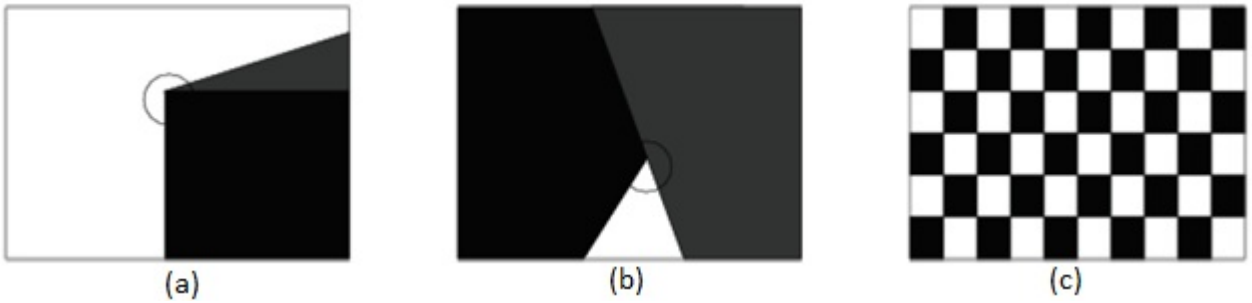

FIGURE 1.1: Exemples illustratifs de trois types de points d'intérêts : (a) coins (b) jonctions en T (c) forte variation d'intensité.

1.5 Type de détecteurs

Il existe une large variété de détecteurs de coins et de points d'intérêt dans la littérature [7]. Seulement quelques uns sont décrits dans cette partie.
De manière générale, un détecteur de points d'intérêts permet de calculer une valeur représentant de l'intérêt pour chaque pixel de l'image. Ensuite, il sélectionne les pixels portant le plus d'informations.
Schmid les a classés dans trois catégories [8] :

- Les détecteurs basés sur les contours.
- Les détecteurs basés sur l'intensité ou le niveau de gris.
- Les détecteurs s'appuyant sur des modèles paramétriques.

1.5.1 Les détecteurs de contours

L'idée est de détecter les contours dans une image dans un premier temps. Les points d'intérêts sont ensuite extraits le long des contours.
La détection des contours est une méthode qui repose sur le calcul des dérivées de la fonction de l'intensité dans l'image : les extrema locaux du gradient de la fonction d'intensité et les passages par zéro du Laplacien [9]. On détecte ainsi les points d'inflexion, où la courbure est localement maximale. On détecte aussi les jonctions présentes sur l'image, qui sont des intersections entre plusieurs contours.
A noter qu'un contour est une frontière entre deux objets dans une image.
Les détecteurs de contours existent depuis longtemps. L'un des plus récents est celui développé par Asada et Brady en 1986 [10]. Cet algorithme extrait des points d'intérêt d'objets en 2D à partir de courbes planes. Ces courbes ont des caractéristiques spécifiques : les variations de courbures. Ces courbures sont classées en plusieurs catégories, les jonctions, les inflexions, etc. Afin d'obtenir une détection robuste, l'application de leur algorithme se fait sur plusieurs échelles.

1.5.2 Les détecteurs utilisant des modèles paramétriques

Les points d'intérêts sont identifiés dans l'image par la mise en correspondance de la fonction d'intensité avec un modèle paramétrique de cette fonction d'intensité.

Ces méthodes s'appuient sur la déformation de ce modèle paramétrique pour qu'il se rapproche des niveaux de gris au voisinage d'un coin. Ces détecteurs sont précis à condition d'avoir de bonnes valeurs initiales pour les paramètres du modèle. Ils fournissent une précision sous-pixellique, mais restent applicables que pour une catégorie spécifique de points d'intérêt. Par exemple, les coins en L. L'un des premiers détecteurs paramétriques a été développé par Rohr en 1992. Il s'agit d'un détecteur de coins en L dont les paramètres sont l'angle entre l'axe de symétrie du coin en L et l'axe des x, les niveaux de gris, la position du point et la quantité de flou [11].
Parida a aussi développé un détecteur général de jonctions. Sa méthode utilise un modèle déformable pour détecter les cloisons radiales [12].
L'une des méthodes que nous étudions et comparons dans ce document fait partie de cette catégorie. Il s'agit du détecteur SFOP.

1.5.3 Les détecteurs exploitant les niveaux de gris

L'idée cette fois-ci est de regarder directement la fonction d'intensité dans les images pour en extraire directement les points de discontinuité. Les approches appartenant à cette catégorie sont celles utilisées généralement pour deux raisons principales. Elles sont plus stables car indépendantes vis-à-vis de la détection des contours. Elles sont aussi plus générales car le type de point d'intérêt n'importe pas. Trois des méthodes que nous étudions dans ce document font partie de cette catégorie. Ce sont les algorithmes SIFT, SURF et ORB.
Il en existe plusieurs dans la littérature. En voici une classification chronologique.

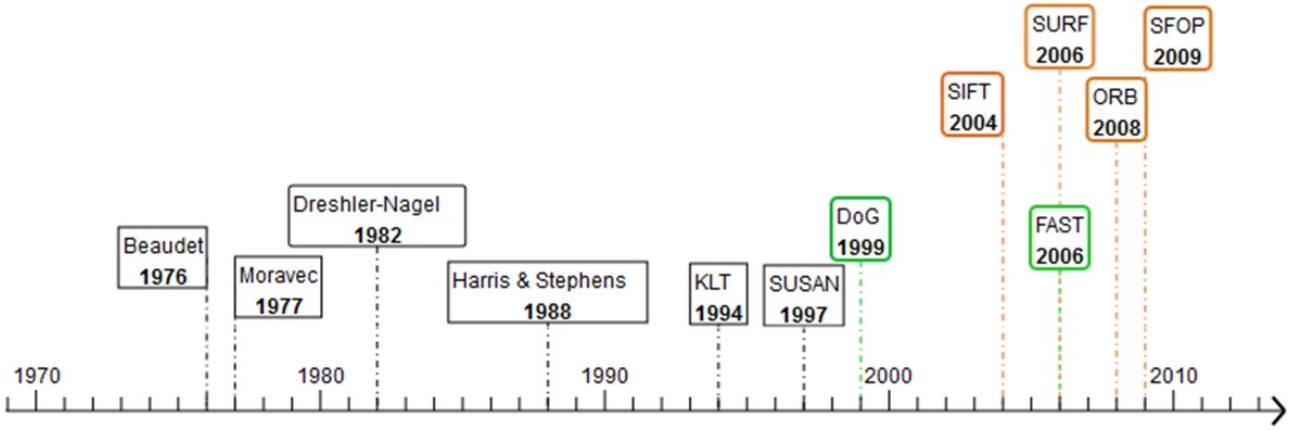

FIGURE 1.2: Frise chronologique

1.5.3.1 Détecteur de Beaudet

Ce détecteur a été développé par Beaudet en 1976. L'idée sur laquelle se base cette méthode est de considérer que des points non-extrema comportent des informations importantes concernant le contour. Il utilise les dérivées partielles secondes.
Théoriquement, l'utilisation de la matrice hessienne permet de déterminer la nature des points critiques d'une fonction. Soit une fonction f à n variables notées (x_i) avec $0 < i < n$. La matrice hessienne est décrite comme suit [13] :

$$\mathbf{H}(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} = \begin{bmatrix} f_{x_1^2} & f_{x_1 x_2} & \cdots & f_{x_1 x_n} \\ f_{x_2 x_1} & f_{x_2^2} & \cdots & f_{x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{x_n x_1} & f_{x_n x_2} & \cdots & f_{x_n^2} \end{bmatrix} \tag{1.1}$$

L'étude des valeurs propres de cette matrice permettent de dire si le point constitue un minimum ou un maximum local, ou encore un point col.
Beaudet lui propose de calculer une matrice proportionnelle à la hessienne décrite comme suit :

$$k(\mathbf{x}) = Cdet(\mathbf{H}(g_\sigma * I(\mathbf{x}))) \tag{1.2}$$

Avec Cconstante.
Les maxima locaux sont définis par :

$$\mathbf{x}_B = argmax_{\mathbf{x}}(|k(\mathbf{x}))|) \tag{1.3}$$

Puis certains critères de sélection ont permis d'éliminer certains points critiques de la fonction. Par exemple, les valeurs propres doivent être du même signe.

1.5.3.2 Détecteur de Dreshler-Nagel

Ce détecteur est une variante du détecteur de Beaudet. Il se base sur la même méthode d'analyse, tout en apportant une amélioration sur la sélection des points d'intérêts [13]. En partant de l'équation 1.2 et en se basant sur le changement de signe de la courbure aux abords d'un coin, il détermine un maximum x_1 puis cherche un minimum local x_2 de k . La ligne rejoignant ces deux extrema permet de déterminer l'endroit où la pente du signal est maximale, c'est-à-dire le point de l'annulation de la courbure :

$$x_{DN} = \frac{x_1 + x_2}{2} \tag{1.4}$$

1.5.3.3 Détecteur de Moravec

L'un des premiers algorithmes de mise en correspondance des images est le détecteur de Moravec. Il se base sur l'extraction des caractéristiques locales de l'image. C'est un détecteur de coins. Il s'agit dans ce cas de comparer des patchs voisins en calculant des sommes des différences au carré. L'algorithme teste chaque pixel pour savoir si un coin est présent.
L'idée de ce détecteur est de considérer le voisinage d'un pixel (une fenêtre) et de déterminer les changements moyens des intensités dans le voisinage considéré lorsque la fenêtre se déplace dans différentes directions. Plus précisément, on considère la fonction suivante :

$$E(x,y) = \sum_{u,v} w(u,v)|I(x+u, y+u) - I(u,v)|^2 \tag{1.5}$$

w : fenêtre considérée (prend la valeur 1 à l'intérieur et 0 à l'extérieur)
$I(u,v)$: Intensité du pixel central
$E(x,y)$: Moyenne du changement d'intensité lorsque la fenêtre est déplacée
Puis, les différents pixels traités sont placés en trois catégories :

1. L'intensité est pratiquement constante dans la fenêtre, et E prend de faibles valeurs.
2. La zone considérée contient un contour rectiligne : La fonction E prendra de faibles valeurs sur le contour, et des valeurs importantes sur les zones perpendiculaires au contour.
3. Si les valeurs de E sont importantes dans toutes les directions, le patch dans ce cas contient un coin ou un point isolé.

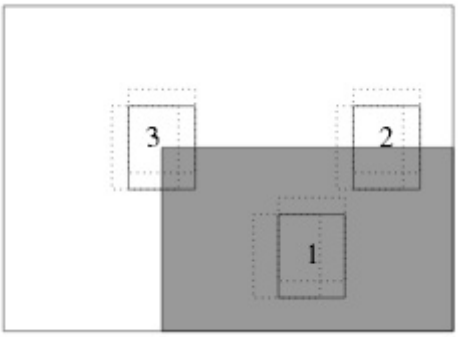

FIGURE 1.3: Les différentes situations considérées

1.5.3.4 Détecteur de Harris

En 1988, Harris et Stephens ont mis au point un détecteur qui améliore celui de Moravec. En effet, ceux-ci ont remarqué que ce détecteur avait deux limitations.

1. Il est premièrement sensible au bruit, car le filtre utilisé est binaire et est appliqué à un voisinage rectangulaire. Ils l'ont remplacé par un filtre gaussien :

$$w(u,v) = exp^{-(u^2+v^2)/2\sigma^2} \tag{1.6}$$

2. Aussi, le détecteur de Moravec est anisotropique (milieu dont les propriétés changent selon les directions) en raison du caractère discret des changements de directions que l'on peut effectuer (des pas de 45° par exemple). Pour améliorer cet aspect, Harris et Stephens ont proposé d'utiliser le développement de Taylos de la fonction d'intensité.

$$I(x+u, y+v) = I(u,v) + x\frac{\partial I}{\partial x} + y\frac{\partial I}{\partial y} + o(x^2, y^2) \tag{1.7}$$

D'où

$$E(x,y) = \sum_{u,v} w(u,v)[x\frac{\partial I}{\partial x} + y\frac{\partial I}{\partial y} + o(x^2, y^2)]^2 \tag{1.8}$$

En négligeant le terme $o(x^2, y^2)$ on obtient la fonction suivante :

$$E(x,y) = Ax^2 + 2Cxy + By^2 \tag{1.9}$$

Avec :

- $A = \frac{\partial I^2}{\partial x} \bigotimes w$
- $B = \frac{\partial I^2}{\partial y} \bigotimes w$
- $C = (\frac{\partial I}{\partial y}\frac{\partial I}{\partial y}) \bigotimes w$

3. Le détecteur de Moravec est aussi fortement sensible aux contours en raison du fait que seul le minimum de E est pris en compte pour chaque pixel. Harris et Stephens ont ainsi mis en place aussi une fonction qui permet de détecter les coins tout en éliminant les bords. Cette fonction prend en compte le comportement général de E. On écrit ainsi :

$$E(x,y) = (x,y).M.(x,y)^T \tag{1.10}$$

Avec

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \tag{1.11}$$

Cette matrice caractérise le comportement local de la fonction E. Les valeurs propres de cette matrice correspondent aux courbures principales associées à E. Un raisonnement similaire à celui de Moravec est appliqué ici :

- Si les deux valeurs propres sont grandes, elles caractérisent un coin.
- Si les deux valeurs propres sont faibles, l'intensité est considérée comme constante.
- Si l'une des valeurs propres est de forte valeur et l'autre est de faible valeur, elles correspondent à un contour.

Le détecteur de Harris a fait ses preuves dans le suivi (tracking) des objets dans une séquence d'images ainsi que dans la reconstruction tridimensionnelle et est utilisé dans d'autres tâches de traitement d'image.

FIGURE 1.4: Exemple de détection de coin à l'aide de Harris

Ces algorithmes sont certes des détecteurs de coins, mais détectent aussi les zones où les différences de gradients sont importantes.

1.5.3.5 Le KLT

Le KLT est un autre détecteur se basant sur celui de Harris. Il a été développé par Shi et Tomasi en 1994. Le KLT signifie Kanade Lucas Tomasi feature tracker. A la différence de Harris et Stephens, le KLT calcule les valeurs propres. Leur idée est de sélectionner des primitives qui peuvent être facilement suivies, lors d'un déplacement de la caméra par exemple. Puis, un pixel est qualifié de point d'intérêt s'il est conforme à des critères qu'ils ont établis. Par exemple, les deux valeurs propres doivent être de fortes valeurs.

1.5.3.6 Détecteur de SUSAN

SUSAN (Smallest Univalue Segment Assimilating Nucleus) est un détecteur de coins développé à l'université d'Oxford en 1997. Ce détecteur utilise un masque en forme de disque centré sur le pixel à analyser.

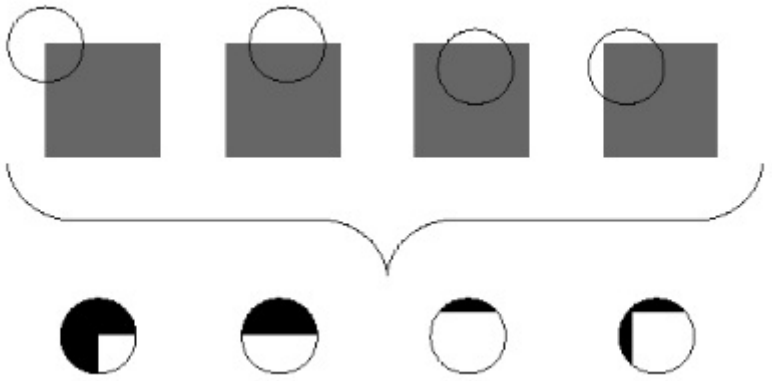

FIGURE 1.5: Exemple du masque d'analyse utilisé par SUSAN (haut) pour le calcul des zones USAN (bas)

Pour chaque pixel P compris dans le masque, on effectue une comparaison avec le pixel central Pn avec la fonction suivante :

$$c(p) = e^{[-I(P)-I(Pn)]^6/t} \tag{1.12}$$

Avec t : rayon du disque
La somme suivante est ensuite calculée :

$$n(Pn) = \sum_{\forall p} c(p) \tag{1.13}$$

n représente la taille de la zone USAN. Cette zone comprend le nombre de pixels de même niveau de gris que P_n. La fonction n est ensuite comparée à un seuil géométrique g afin d'obtenir la fonction R suivante :

$$R(P_n) = \begin{cases} g - n(P_n) & Si : n(P_n) < g \\ 0 & Sinon \end{cases} \tag{1.14}$$

g est fixé à $\frac{n_{max}}{2}$ avec n_{max} est la taille maximale de la zone USAN. Ceci revient à dire que si le centre du cercle d'analyse se trouve sur un coin, alors la zone USAN sera strictement plus petite que la demi-surface du masque.

1.5.3.7 Détecteur basé sur les différences de Gaussiennes (DoG)

Afin de rendre plus stable la détection des points d'intérêts, Lowe propose en 1999 un nouveau détecteur. Celui-ci construit un espace échelle afin de calculer les différences de Gaussiennes.
La construction d'un espace échelle se fait à travers des convolutions consécutives de l'image avec un filtre Gaussien, en augmentant la valeur de l'échelle à chaque fois.

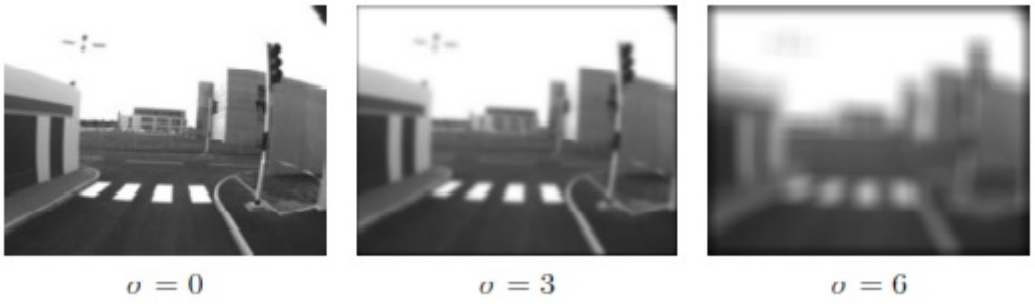

FIGURE 1.6: Représentation multi-échelle d'une image

L'idée est donc de supprimer progressivement des détails de l'image afin de simuler d'éventuels changements d'échelle.
Cette convolution s'exprime comme suit :

$$L(x, y; \sigma) = g_\sigma * I(x, y) \tag{1.15}$$

Ainsi, Lowe définit la fonction $D(x, y; \sigma)$ comme suit :

$$D(x, y; \sigma) = L(x, y; k\sigma) - L(x, y; \sigma) \tag{1.16}$$

Avec k facteur multiplicateur constant.

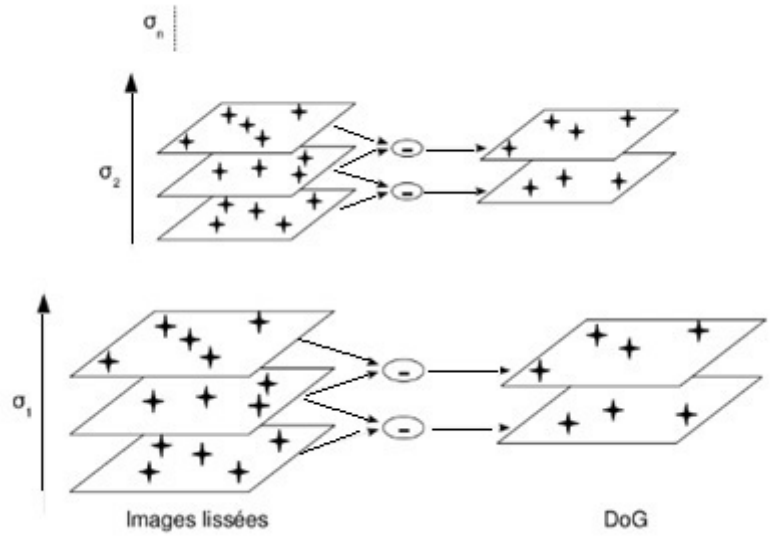

FIGURE 1.7: Illustration de l'octave des images lissées et de celle des différences de Gaussiennes

L'octave des différences de Gaussiennes permet de déterminer les extrema locaux. Puis, Lowe apporte quelques précisions à cette méthode dans un nouvel algorithme qu'il appellera SIFT. Ce dernier est utilisé dans le cadre de ce mémoire et sera détaillé dans le chapitre qui suit.

1.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'un des grands domaines de la vision par ordinateur : La reconnaissance d'objets. Nous avons ainsi présenté les défis majeurs qui régissent ce domaine. Nous avons aussi vu les différentes approches suivies lors de la reconnaissance d'un objet sur une image en mettant l'accent sur les méthodes qui se basent sur les points d'intérêt pour caractériser les images. C'est la catégorie à laquelle appartiennent les méthodes étudiées dans ce mémoire. Enfin, nous avons établi un historique de ces méthodes.

Chapitre 2

Présentation des méthodes étudiées : SIFT, SURF,ORB,SFOP

Introduction

Dans le chapitre précédent, nous avons vu un historique de l'évolution des méthodes locales de reconnaissance d'objet. Nous avons surtout mis l'accent sur les algorithmes se basant sur la détection des points d'intérêt.
Dans ce chapitre, nous exposons l'aspect théorique des quatre méthodes étudiées. Nous détaillons pour chacune (SIFT, SURF, ORB et SFOP) les différentes étapes : de la détection des points clés jusqu'au calcul des descripteurs.

Notons que dans tout ce qui suit, l'image est convertie en niveau de gris avant tout traitement

2.1 SIFT

Le SIFT (Scale-Invariant Feature Transform), qui peut être traduit par : Caractérisation d'images par descripteurs locaux invariants à l'échelle, est un algorithme de traitement d'images. Il permet de détecter les similarités entre deux images numériques independemment de l'echelle.
Cette approche permet la détection et l'extraction des descripteurs de caractéristiques locales qui sont assez invariants aux changements d'illumination, d'échelle, de rotation,

au bruit de l'image et aux petits changements d'angles lors de la prise d'image. Cet algorithme a été proposé par David Lowe en 1999 [14] , qui l'a amélioré ensuite 2004 [15].
La méthode consiste en trois étapes fondamentales : la détection des points d'intérêts, le calcul des descripteurs et enfin la correspondance entre images.

2.1.1 Détection des points d'intérêts

2.1.1.1 Construction de l'espace-échelle Gaussien

Les points d'intérêts SIFT correspondent aux extrema locaux des différences de filtres Gaussiens à différentes échelles.

La construction de cet espace de différences de Gaussiennes se fait en deux temps. Tout d'abord l'image est convoluée à plusieurs filtres Gaussiens, comme suit :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.1}$$

Avec
$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$
$I(x, y)$: L'image en niveaux de gris.
x, y : Coordonnées des pixels de l'image.
σ : Ecart-type du filtre Gaussien.
Chaque filtre est caractérisé par son propre facteur d'échelle σ.
Le facteur d'échelle est fixé à une valeur initiale σ_0 , et augmenté à chaque fois. Plus exactement, l'échelle est multipliée par un facteur k. Ce facteur est déterminé par le nombre s d'images qu'on veut obtenir par octave dans l'espace-échelle Gaussien suivant la relation $k = 2^{1/s}$.
Lorsque la valeur $2\sigma_0$ est atteinte :

- Les images filtrées jusqu'à présent constituent une octave.
- Les dimensions de l'image sont réduites de moitiés, et l'algorithme est reproduit de nouveau avec cette image réduite pour obtenir une seconde octave.
- Le calcul des images filtrées est arrêté lorsque les dimensions de l'image deviennent très petites.

A la fin de cette étape nous obtenons un ensemble d'images floues à des échelles différentes, comme illustrées a gauche de la figure 2.1.

Une fois, les octaves obtenues, l'étape suivante consiste à calculer les différences entre deux images successives :

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.2}$$

L'ensemble des différences de Gaussienne $D(x, y, \sigma)$ constitue l'espace-échelle Gaussien. Comme illustré à droite de la figure 2.1.
D. Lowe recommande la valeur $\sigma_0 = 1.6 * k$.[15]
Nous devons produire $S = s + 3$ images dans la pile d'images floues pour chaque octave, de sorte que la détection des extrema, abordée au paragraphe suivant puisse couvrir une octave complète.

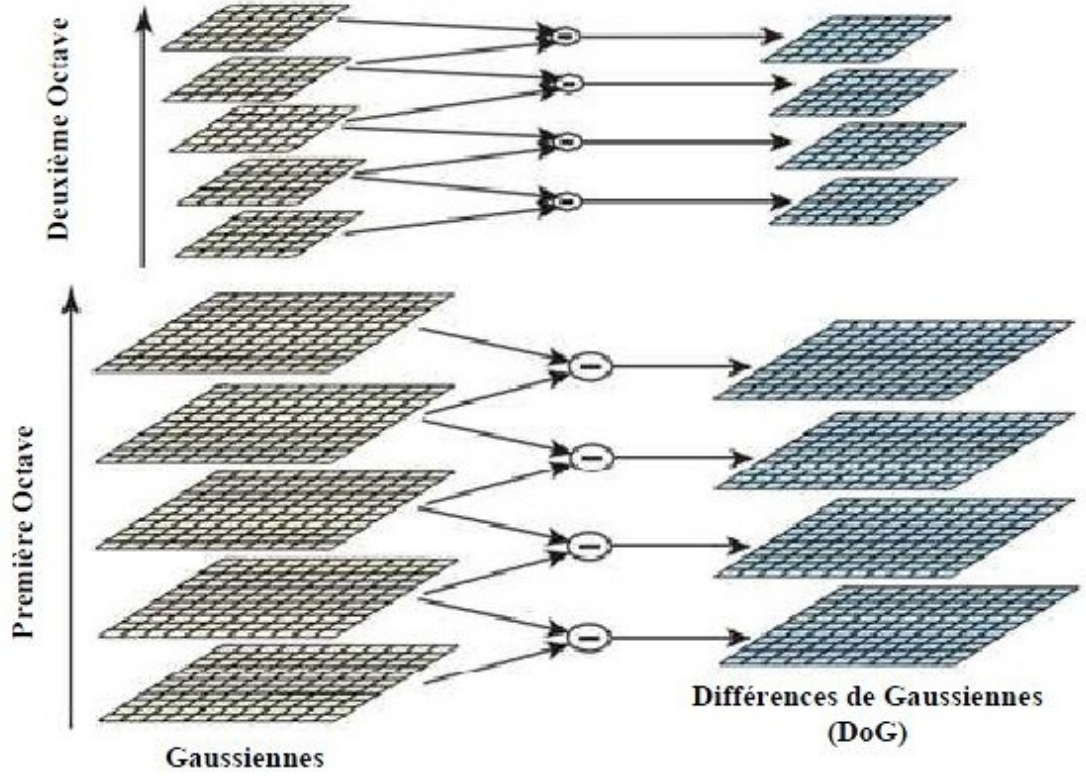

FIGURE 2.1: Illustration de l'espace-échelle Gaussiennes et différence de Gaussiennes

2.1.1.2 Localisation des extrema locaux

Une fois l'espace des différences de Gaussiennes obtenu, il suffit de calculer les extrema locaux. Ces extrema sont sélectionnés de la manière suivante : Chaque pixel est comparé avec ses 26 voisins (8 pixels voisins sur la même image, 9 pixels sur chacune des images au dessus et en dessous de son image dans l'espace-échelle Gaussien)comme illustré sur

la figure 2.2.
Le pixel est sélectionné seulement si c'est le maximum ou le minimum de tous ses voisins.

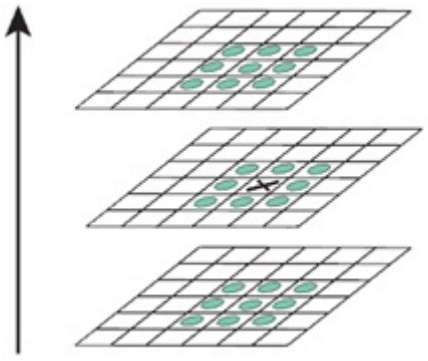

FIGURE 2.2: Localisation des extrema locaux

Ces extrema sont des points clés potentiels. Les étapes suivantes vont permettre de les relocaliser avec précision et d'en éliminer un certain nombre.

2.1.1.3 Amélioration de la précision par interpolation des coordonnées

Lorsqu'un point d'intérêt est localisé sur un étage de l'espace échelle, différent du premier étage, une optimisation sous-pixelique est effectuée, afin de positionner au mieux ce pixel sur l'image de taille initiale.
Cela s'obtient par un développement de Taylor d'ordre 2 de la fonction $D(x, y, \sigma)$, en prenant comme origine les coordonnées du point d'intérêt candidat :

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x^T}\frac{\partial^2 D^T}{\partial \mathbf{x}^2}\mathbf{x} \tag{2.3}$$

où $\mathbf{x} = (x, y, \sigma)^T$au voisinage du point d'intérêt——
La position précise de l'extremum $\hat{\mathbf{x}}$ est déterminée en résolvant l'équation annulant la dérivée de cette fonction par rapport à $\mathbf{x}$:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}} \tag{2.4}$$

Si le $\hat{\mathbf{x}} > 0.5$ dans l'une des trois dimensions, cela signifie que le point est plus proche d'un des voisins dans l'espace des échelles discret.
Dans ce cas, le point d'intérêt candidat est mis à jour et l'interpolation est réalisée à partir des nouvelles coordonnées. Sinon, $\hat{\mathbf{x}}$ est ajouté au point candidat initial qui gagne ainsi en précision.

2.1.1.4 Élimination des points d'intérêts de faible contraste

La valeur de la fonction $D(\hat{\mathbf{x}})$ est utile pour l'élimination d'extrema avec faible contraste. En combinant les deux équations précédentes nous trouvons :

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\hat{\mathbf{x}} \tag{2.5}$$

Un seuillage absolu ($|D(\hat{\mathbf{x}})| < 0.03$) est effectué pour éliminer les points instables, à faible contraste.

2.1.1.5 Élimination des points situés sur les arêtes

Les points situés sur les arêtes (ou contours) doivent être éliminés. Ceci, car la fonction $D(x, y, \sigma)$ y prend des valeurs élevées, ce qui donne naissance à des extrema locaux instables, très sensibles au bruit.
Un point instable aura une grande courbure le long du contour, mais une faible courbure dans la direction perpendiculaire. Afin d'éliminer ces points, une matrice Hessienne 2x2 est calculée à la position et échelle du point clé :

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{2.6}$$

Les dérivées partielles sont estimées en effectuant les différences des voisions du point clé. Les valeurs propres de la matrice Hessienne sont proportionnelles aux courbures principales de D.
Mais comme nous nous intéressons uniquement au rapport des deux valeurs propres, il est inutile de les calculer, puisque la trace et le déterminant de la matrice permettent de déduire respectivement la somme et le produit de ces deux valeurs.

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta \tag{2.7}$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \tag{2.8}$$

En supposant que α est la plus grande valeur propre et r le rapport entre la plus grande et la plus petite valeur propre $r = \alpha/\beta$
Nous avons :

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{2.9}$$

Comme la fonction $\frac{(r+1)^2}{r}$ est strictement croissante sur $[1, +\infty]$, donc pour vérifier que le rapport r est au dessus d'un certain seuil r_{seuil}, il suffit de vérifier que :

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r_{seuil}+1)^2}{r_{seuil}} \tag{2.10}$$

Ceci est beaucoup moins coûteux en nombre d'opérations, que le calcul des valeurs propres.
Lowe recommande de fixer r_{seuil} à 10 , et donc d'éliminer les points clés où le rapport des deux principales courbures est supérieur à 10 [15].

2.1.2 Calcul des descripteurs

2.1.2.1 Assignation d'orientation

Maintenant que les points clés sont déterminés, la présente étape est la dernière avant le calcul des descripteurs. Elle permet d'attribuer à chacun une ou plusieurs orientations déterminées localement sur l'image. C'est ce qui assurera l'invariance de la méthode part rapport à la rotation et au changement d'échelle.

Pour un point clé donné (x_0, y_0, σ_0), nous calculons d'abord la norme $m(x, y)$ et l'orientation $\theta(x, y)$. Le calcul s'effectue au niveau de ses points voisins sur l'image filtrée, calculée au départ $L(x, y, \sigma)$ avec σ_0 le plus proche du facteur d'échelle du point :

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{2.11}$$

$$\theta(x,y) = tan^{-1}\frac{(L(x+1,y) - L(x-1,y))}{(L(x,y+1) - L(x,y-1))} \tag{2.12}$$

Une fois ce calcul préliminaire effectué, un histogramme des orientations est réalisé avec des intervalles couvrant chacun 10 degrés d'angle(figure 2.3). L'histogramme est doublement pondéré : d'une part, par une fenêtre circulaire Gaussienne de paramètre égal à $1.5\sigma_0$, d'autre part, par l'amplitude de chaque point.
Les pics dans cet histogramme correspondent aux orientations dominantes. Toutes les orientations dominantes permettant d'atteindre au moins 80% de la valeur maximale sont prises en considération. Ce qui provoque si nécessaire la création de points-clés supplémentaires ne différant que par leur orientation principale.
À l'issue de cette étape, un point-clé est donc défini par quatre paramètres(x, y, σ, θ).

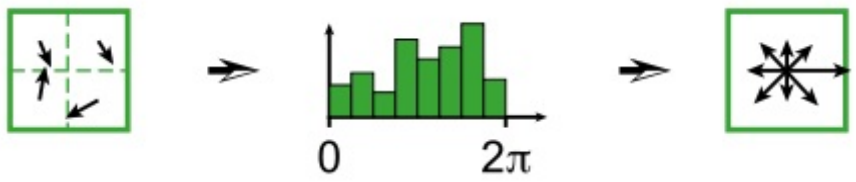

FIGURE 2.3: Construction de l'histogramme des orientations d'un point clé

2.1.2.2 Descripteur SIFT du point d'intérêt

Après avoir désigné les points clés et leur orientation principale, il est maintenant temps de calculer le vecteur descripteur de chaque point clé. Tout comme l'étape précédente, le calcul qui suit s'effectue sur l'image lissée $L(x, y, \sigma)$ avec σ le plus proche du facteur d'échelle du point.

Pour chaque point, on commence par modifier le système de coordonnées local, en utilisant une rotation d'angle égal à l'orientation du point-clé, mais de sens opposé. On considère ensuite, toujours autour du point-clé, une région de 16×16 pixels, subdivisée en 4×4 zones de 4×4 pixels chacune. Sur chaque zone est calculé un histogramme des orientations comportant 8 intervalles.

En chaque point de la zone, l'orientation et l'amplitude du gradient sont calculés comme précédemment. L'orientation détermine l'intervalle à incrémenter dans l'histogramme, ce qui se fait avec, comme précédemment, une double pondération :Par l'amplitude et par une fenêtre Gaussienne centrée sur le point clé, de paramètre égal à $0,5$ fois le facteur d'échelle du point-clé comme l'illustre la figure 2.4.

Ensuite, les 16 histogrammes à 8 intervalles chacun sont concaténés et normalisés. Dans le but de diminuer la sensibilité du descripteur aux changements de luminosité, les valeurs sont plafonnées à $0,2$ et l'histogramme est de nouveau normalisé, pour finalement fournir le descripteur SIFT du point-clé, de dimension 128.

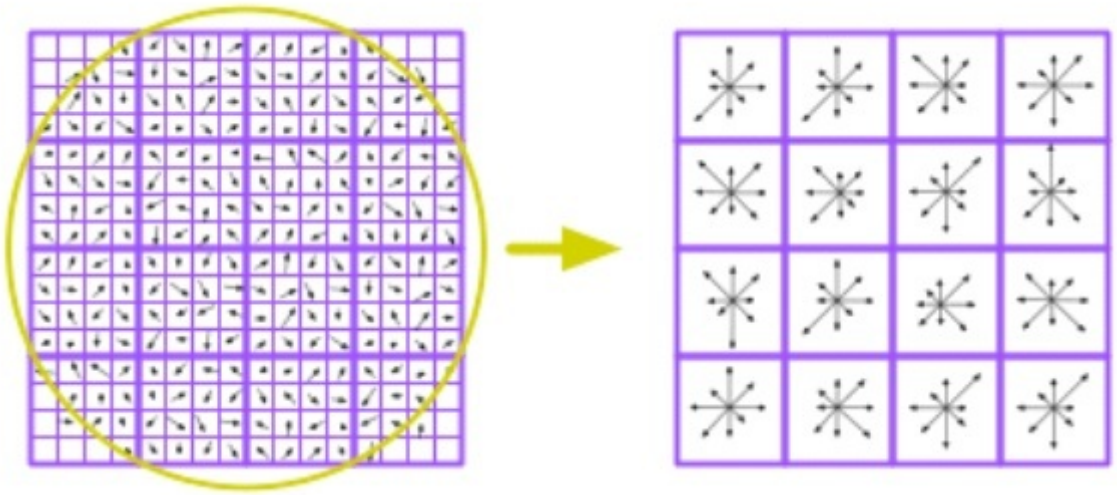

FIGURE 2.4: Calcul du descripteur d'un point clé

2.1.3 Correspondance entre images

La problématique de base pour laquelle la méthode SIFT a été conçue est la suivante : peut-on trouver dans une image donnée (image requête), des objets déjà présents dans une collection d'images de référence préétablie ?
Afin de parvenir à cet objectif, il faut extraire de chaque image de la collection ces points clés et stocker les descripteurs. Ainsi au moment de la comparaison, il faut pour chaque point clé de l'image requête déterminer son plus proche voisin, en utilisant la distance Euclidienne, sur chaque image de la collection.
Pour ce faire, Lowe utilise un algorithme d'approximation " Best-Bin-First "(BBF) afin d'éviter une recherche exhaustive.

2.2 SURF

2.2.1 Introduction

SURF ou Speeded Up Robust Features, qui signifie caractéristiques robustes accélérées. La recherche des correspondances des images discrètes peut être divisée en trois étapes principales.
Tout d'abord, Il faut sélectionner les points d'intérêt, à différents emplacements sur l'image, comme les coins, les taches et les jonctions en T. L'étape est répétée plusieurs fois afin de tester sa répétabilité, c'est-à-dire, si les mêmes points d'intérêt sont détectés

sous des conditions de vision différentes. Puis, le voisinage de chaque point d'intérêt est représenté par un vecteur qu'on appellera descripteur. Ce descripteur doit être distinctif et robuste vis-à-vis du bruit, de la détection d'erreurs ainsi qu'aux déformations géométriques ou photométriques.
A la fin, les vecteurs descripteurs sont mis en correspondance entre différentes images. Cette étape est souvent basée sur la distance entre les vecteurs, la distance Euclidienne par exemple. On utilise aussi la distance de Mahalanobis, qui mesure la corrélation et la similarité entre deux variables.
Le temps de calcul est directement lié aux dimensions des descripteurs calculés.
Le but principal de cette méthode, a été le développement d'un détecteur et d'un descripteur en même temps, qui sont plus rapides à calculer en comparaison avec les méthodes développées précédemment. Toutefois, les performances n'ont pas été sacrifiées. D'ailleurs, l'étude comparative (Juan et Al., 2010) démontre la supériorité du descripteur SURF par rapport à SIFT d'un point de vue de ses performances en temps d'exécution et de sa robustesse aux changements d'illumination.

2.2.2 Détermination des descripteurs

2.2.2.1 Détecteur SURF

Le détecteur SURF est basé sur le détecteur de points Fast Hessian, mais en utilisant une approximation très basique. Ce détecteur utilise les images intégrales afin de réduire le temps de calcul, d'où l'appellation 'Fast Hessian Detector '.

2.2.2.1.1 Image intégrale Une image intégrale, est un algorithme qui permet de représenter une image sous format numérique, permettant de calculer rapidement des sommes de valeurs dans des zones rectangulaires.
Cette représentation est une image de même taille que l'originale, dont chacun des points contient la somme des pixels situés au-dessus et à gauche de ce point. Cet algorithme est défini par l'expression suivante :

$$ii(x,y) = \sum_{x' \leqslant x, y' \leqslant y} i(x',y') \tag{2.13}$$

Grâce à cette méthode, la somme des valeurs dans une zone rectangulaire peut être calculée avec seulement 4 accès de l'image intégrale (6 accès pour deux zones rectangulaires contiguës), et donc un temps constant quelle que soit la taille de la zone.

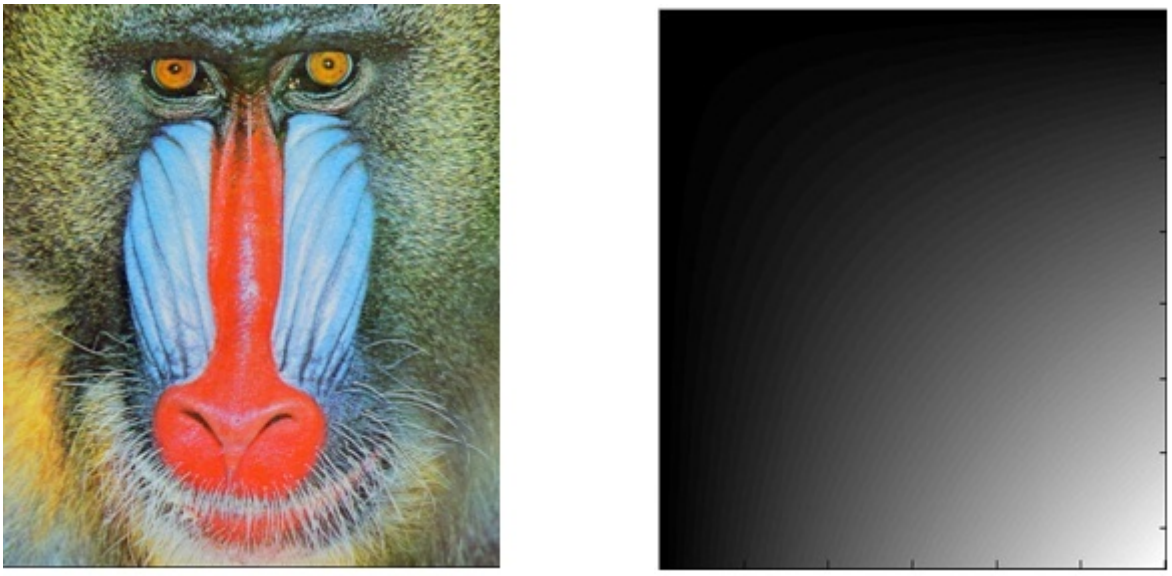

FIGURE 2.5: Une image et son image intégrale

2.2.2.1.2 Détecteur Fast Hessian

La matrice Hessienne est utilisée dans cette méthode pour ses performances en termes de rapidité et de précision.
Dans le détecteur de 'Hessian-Laplace', il fallait deux opérations distinctes pour la localisation et la détermination de l'échelle. Le détecteur Fast-Hessian se base sur le déterminant de la matrice pour effectuer les deux opérations.
Les zones de fort changement d'intensité des pixels sont recherchées dans l'image.
La matrice Hessienne basée sur le calcul des dérivées partielles d'ordre deux, est utilisée pour ça.

Un point donné $\mathbf{x} = (x, u)$ dans une image I est défini par sa matrice Hessienne $\mathbf{H}(\mathbf{x}, \sigma)$ au point $\mathbf{x}$ et à l'echelle σ . La matrice Hessienne est définie comme suit :

$$\mathbf{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \tag{2.14}$$

Où : $L_{xx}(\mathbf{x}, \sigma)$ est la convolution de la dérivée Gaussienne de second ordre 2 $\frac{\partial^2 g(\sigma)}{\partial \mathbf{x}}$ avec l'image I au point $\mathbf{x}$.
Afin de gagner en rapidité, les Gaussiennes sont approximées par des fonctions à paliers appelées box filter. Un box filter est une simple moyenne des pixels d'une région donnée. Si la région est de taille 7x7, le résultat du box filter est la moyenne des intensités de 49 pixels.
En effet,convoluer une image plusieurs fois avec un box filter, s'approximerait au traitement avec le filtre de Gauss.

Il a évidemment été prouvé que les performances des box filters sont tout à fait comparables à celles données par les dérivées Gaussienne originales.
Les éléments de la matrice Hessienne ont été renommés de la manière suivante : D_{xx}, D_{xy} et D_{xy}.
Aussi, l'application des box filters ne se fait pas en série, mais en parallèle. C'est-à-dire que nous n'avons pas besoin de la sortie du premier filtre pour lui appliquer le deuxième, mais le traitement se fait sur l'image intégrale.
Le premier masque appliqué a une taille de 9x9 pixels. Il correspond à l'échelle$s = 1.2$ (ce qui correspond à $\sigma = 1.2$ pour le filtre Gaussien). De plus grands masques sont ensuite appliqués : 15x15, 21x21,27x27 etc.
Il existe une certaine cohérence entre le choix des masques et les échelles. Ainsi, un masque de 27x27 correspond à une échelle $s = 3$x1.2.
L'application des différents box filters nous permettra de construire une pyramide.
On a pu aussi établir une nouvelle expression du déterminant de la matrice approximée :

$$Det(H_{approximee}) = D_{xx} * D_{yy} - (0.9D_{xy})^2 \tag{2.15}$$

Si le déterminant de la matrice Hessienne est positif, alors les valeurs propres de la matrice sont toutes les deux positives ou toutes les deux négatives. Ceci signifie qu'un extremum est présent. Les points d'intérêt sont après localisés là où le déterminant est maximal, dans un voisinage de 3x3x3, comme le montre la figure 2.6

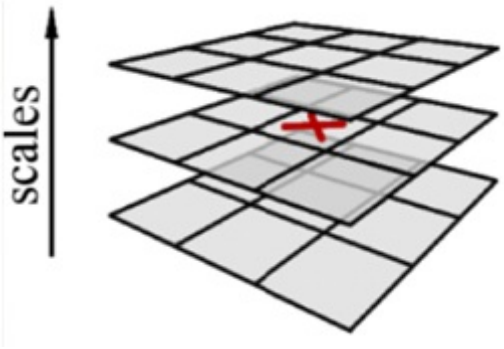

FIGURE 2.6: Localisation des points d'intérêt dans la pyramide

2.2.2.2 Descripteurs SURF

Une fois les points d'intérêts extraits, la seconde étape du SURF consiste à calculer les descripteurs correspondants. Cette étape elle-même consiste en deux opérations.

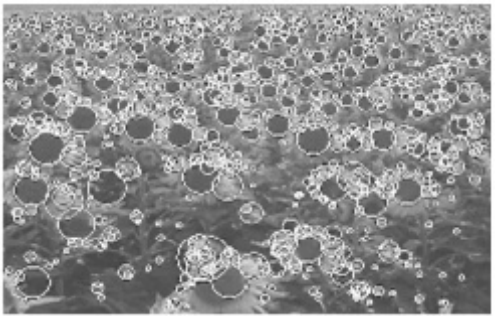

FIGURE 2.7: Exemple de détection des points d'intérêt avec le détecteur Fast Hessian

2.2.2.2.1 Assignation des orientations Afin d'être invariant par rapport à la rotation, on identifie l'orientation dominante de chaque point d'intérêt. Le descripteur SURF décrit l'intensité des pixels dans un voisinage autour de chaque point d'intérêt. La réponse en x et en y des ondelettes de Haar est calculée dans un voisinage de $6 * \sigma$ où σ est l'échelle à laquelle le point d'intérêt a été trouvé

Ondelette de Haar

Dans cet algorithme, on parle d'ondelette de Haar, mais en vérité, il s'agit de l'utilisation des caractéristiques pseudo-Haar. Elles tiennent leur nom de la similarité avec l'ondelette de Haar. La technique qui emploie ces caractéristiques consiste à définir des zones rectangulaires et adjacentes sur l'image. On calcul ensuite la somme des intensités des pixels de l'image dans ces zones. La différence entre les rectangles noirs et blancs donne la caractéristique pseudo-Haar

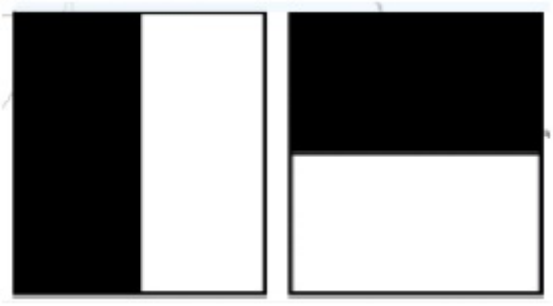

FIGURE 2.8: Ondelettes de Haar pour calculer les réponses en x (à gauche) et les réponses en y (à droite).

Les réponses obtenues en x et en y sont représentées par des vecteurs. A partir des valeurs obtenues, l'orientation dominante de chaque point d'intérêt est calculée de la manière suivante : On fait glisser une fenêtre d'orientation qui recouvre un angle de $\frac{\pi}{3}$, et on calcule la somme de toutes les réponses comprises dans cette fenêtre.

Les deux sommes, résultant de la somme des réponses en **x** et de celle des réponses en **y**

, génèrent un nouveau vecteur.
Le vecteur le plus long, représentera l'orientation dominante.

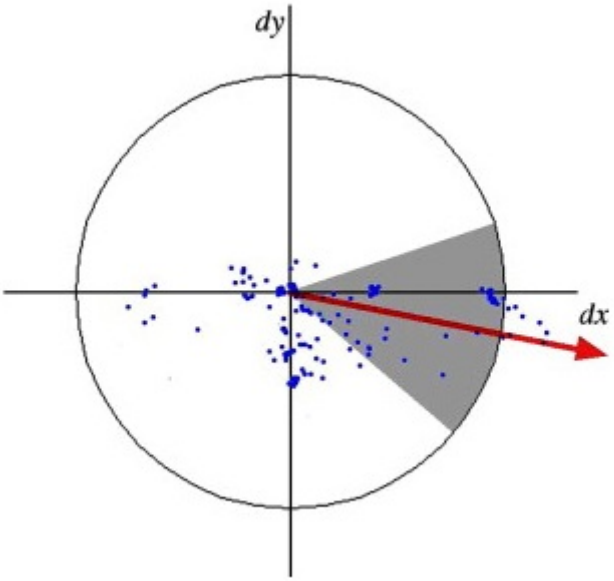

FIGURE 2.9: La fenêtre glissante utilisée pour l'assignation des orientations

2.2.2.2.2 Composants du descripteur La première étape de l'extraction des descripteurs, consiste en une construction d'une région carrée autour de chaque point d'intérêt, qui constituera évidemment le centre. Ce carré sera orienté selon l'orientation dominante calculée précédemment. La taille de cette fenêtre est de 20σ. Un exemple est illustré sur la figure 2.10.

FIGURE 2.10: Détails d'un graffiti, présentant les tailles des fenêtres carrées à différentes échelles

Chaque région est subdivisée en $4 * 4$ sous régions. Pour chacune de ces sous régions, les ondelettes de Haar sont calculées sur$5 * 5$ points. On appellera dx la réponse horizontale à l'ondelette de Haar, et dy la réponse verticale. Ces réponses sont pondérées à l'aide d'un filtre Gaussien centré au point d'intérêt.

La somme des réponses verticales et horizontales de chaque sous-région représentera la première partie des caractéristiques du point d'intérêt. Puis, afin d'avoir plus d'informations concernant les changements d'intensité, nous calculerons aussi la sommes des valeurs absolues $|dx|$et $|dy|$.Cette étape est illustrée sur la figure 2.11. Ainsi, chaque sous-région

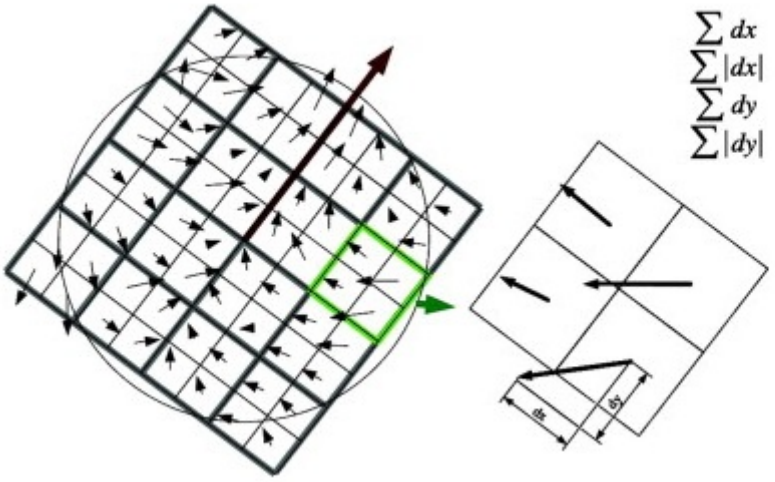

FIGURE 2.11: Calcul des 4 caractéristiques d'une sous-région. Ici, la sous-région est subdivisée en 2x2 pour un but pédagogique

est décrite par 4 caractéristiques. Par conséquent, tout point d'intérêt est représenté par un vecteur de 64 composantes$(4 * 4 * 4)$ La figure 2.12 représente les propriétés de 3 sous-régions, avec des intensités différentes. Ainsi, sur une région homogène comme celle

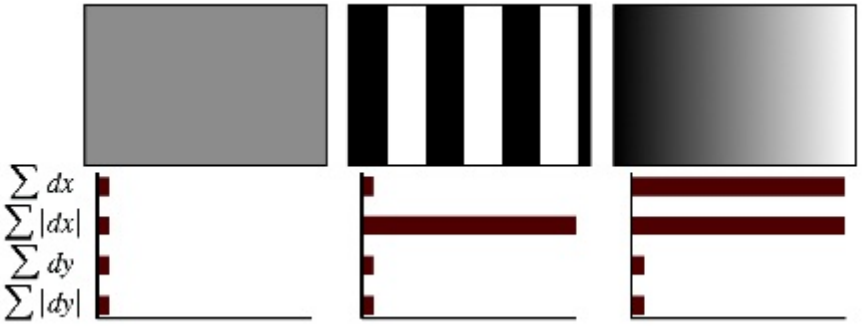

FIGURE 2.12: Comparaison des caractéristiques de 3 sous-régions d'intensités différentes

représentée à gauche, toutes les valeurs sont relativement basses. Celle du milieu présente plusieurs fréquences sur $\mathbf{x}$, et donc la valeur $|dx|$ est la seule élevée. Si l'intensité

augmente graduellement comme sur la région de droite, dx et $|dx|$ sont toutes les deux élevées.

2.3 ORB

2.3.1 Introduction

ORB est l'acronyme de Oriented Rotated Brief. Cela signifie Descripteur BRIEF orienté. Le SIFT, a fait des preuves remarquables ces dix dernières années dans le domaine de la vision par ordinateur. Il ne convient malheureusement pas pour les applications en temps réel, à cause de sa gourmandise en temps d'exécution. Une recherche intensive a été menée pour le remplacer. Le SURF en est le meilleur résultat. Il n'est pourtant pas aussi performant que le SIFT. La méthode ORB présentée dans cette partie allie rapidité et performance.
L'ORB est une méthode extrêmement rapide est basée sur les descripteurs binaires BRIEF(Binary Robust Independent Elementary Features)[16] et sur le détecteur de coins FAST(Features from Accelerated Segment Test). Elle est invariante à la rotation et résistante au bruit. Les tests réalisés par les auteurs ont prouvé que cet algorithme était deux fois plus rapide que le SIFT, présenté plus haut, bien qu'il présente des performances tout aussi bonnes[17]. Son efficacité a été testée sur plusieurs applications dans le monde réel, notamment la détection des objets ainsi que les applications de tracking sur des smartphones.

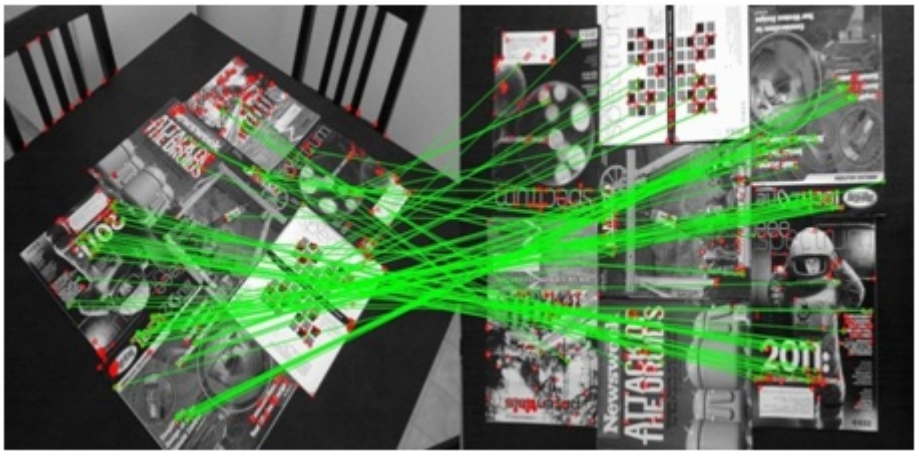

FIGURE 2.13: Exemple de l'application de ORB dans le monde réel. Détection des objets sous différents point de vue. Les lignes vertes représentent les correspondances valides, les cercles rouges les points non matchés

2.3.2 Détection des points clés

2.3.2.1 FAST : Détecteur de coins

Pour la détection des points clés, ORB se base sur le détecteur de coins FAST (Features from Accelerated Segment Test). Ce détecteur construit un cercle d'un rayon de trois pixels autour d'un pixel central pour effectuer le calcul. Prenons par exemple les pixels suivants, donc P est le pixel central.

Un vecteur de 16 pixels est ainsi obtenu. Une fois ce vecteur en main, les différences

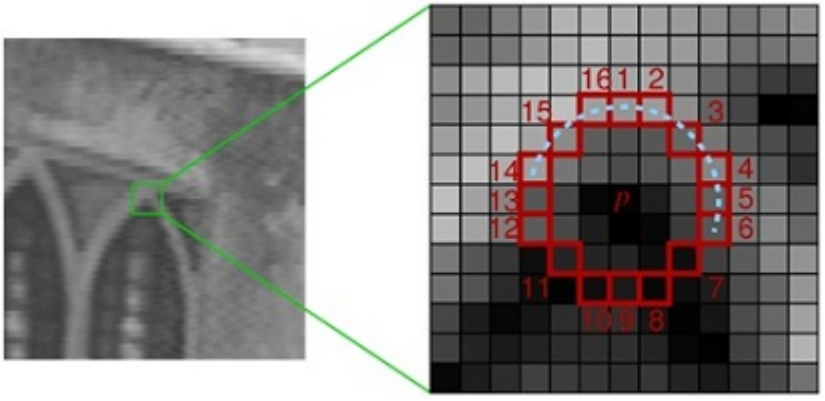

FIGURE 2.14: Fonctionnement du détecteur FAST

d'intensité entre les pixels du vecteur et le vecteur central sont calculées.Un seuil haut et un seuil bas sont ensuite choisis autour de la valeur du pixel central. Ainsi, trois catégories de pixels sont définies :

- -1 si le résultat est inférieur au seuil bas
- 0 si le résultat est entre le seuil bas et le seuil haut
- $+1$ si le résultat est supérieur au seuil haut

Afin de comprendre l'algorithme, prenons l'exemple de la figure 2.15 : En appliquant l'algorithme précédent, nous obtenons le vecteur suivant :

$$[0000011111111111]$$

Nous pouvons remarquer que pour détecter un coin, une succesion de onze 1 doit etre contenue dans le vecteur. Il en va de meme pour une suite de onze -1. Afin de rendre la détection plus flexible, cet algorithme propose une détection de suites de 10 à 12 points. Celles-ci ont donné de bons résultats.

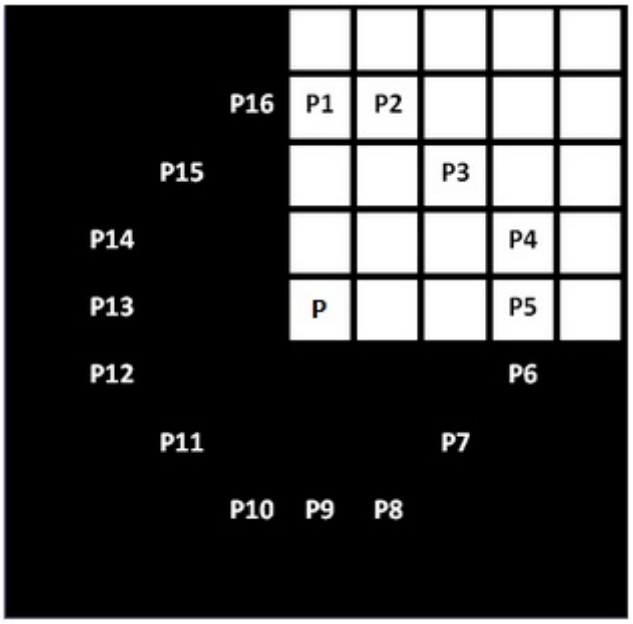

FIGURE 2.15: Exemple explicatif FAST

Afin de minimiser le nombre d'opérations à effectuer, le test se fait d'abord sur les pixels $P1$, $P5$, $P9$ et $P13$. Il est évident que si le pixel $P5$ est dans la catégorie 0, il n'est pas possible d'obtenir une suite de 1 ou de -1 entre $P1$ et $P12$.
La figure 2.16 montre le résultat obtenu après le traitement FAST sur Matlab.

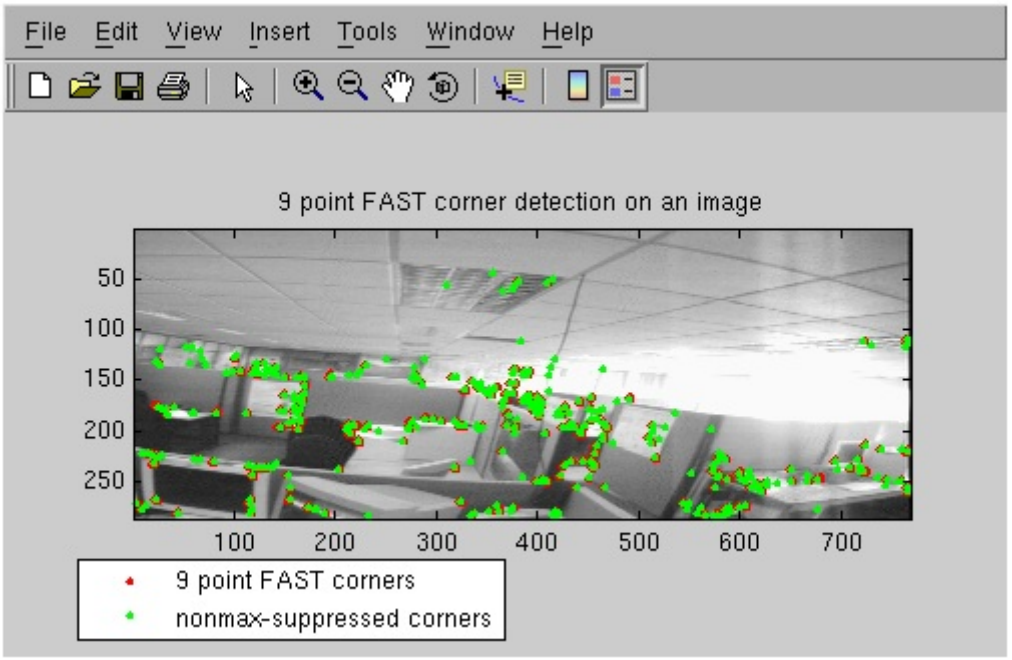

FIGURE 2.16: Détection des coins avec FAST

2.3.2.2 FAST et ORB

Comme nous l'avons vu auparavant, le détecteur FAST ne se base pas sur l'échelle pour détecter les points d'intérêt. Pour cela, ORB construit un espace échelle. C'est-à-dire que le FAST est appliqué à une pyramide obtenue en augmentant l'échelle de l'image.
Afin d'avoir une détection de points d'intérêt plus précise, la condition du détecteur de Harris est appliquée aux points élus par FAST. C'est-à-dire,qu'une matrice de Harris est calculée en tout point. Puis, selon les valeurs propres de celles-ci, une élection des N meilleurs est effectuée :

1. Si les deux valeurs propres sont pratiquement nulles, le point n'a aucun intérêt.
2. Si l'une des valeurs propres est nulle et l'autre est grande et positive, le point corresponderait à un bord.
3. Si les deux valeurs propres sont grandes et positives, le point correspond à un coin.

2.3.2.3 Descripteur BRIEF

Le descripteur BRIEF est un descripteur binaire qui effectue de simples tests entre les pixels d'un patch dans une image floutée. Il a des performances similaires au SIFT, notamment sa robustesse par rapport à l'échelle et la luminosité. Cependant, il affiche des performances médiocres quant à la robustesse par rapport à la rotation[16].
Considérons un patch P, d'une image floutée. Un test est effectué sur l'intensité des pixels du patch :

$$\tau(\mathbf{p};\mathbf{x},\mathbf{y}) := \begin{cases} 1 & : \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & : \mathbf{p}(\mathbf{x}) \geqslant \mathbf{p}(\mathbf{y}) \end{cases} \tag{2.16}$$

Où : $p(x)$ est l'intensité de P au point x.
La caractéristique est définie sur n tests binaires comme suit :

$$f_n(\mathbf{p}) := \sum_{1 \leqslant i \leqslant n} 2^{i-1} \tau(\mathbf{p};\mathbf{x}_i,\mathbf{y}_i) \tag{2.17}$$

Plusieurs distributions ont été considérées dans [16]. Mais celle qui donne l'une des meilleures performances est la distribution Gaussienne. Le choix s'est naturellement porté dessus dans BRIEF. La taille du vecteur a aussi été fixée à n=256.
Flouter l'image est très important dans cette procédure. L'image intégrale est d'ailleurs utilisée sur des fenêtres de 5x5 patchs de 31x31 pixcels. Le choix a été fait après multiples expérimentations[17].

2.3.2.4 Steered BRIEF ou BRIEF orientés

Comme il a été mentionné plus haut, les descripteurs BRIEF ne sont pas invariants par rapport à la rotation. Une étape de traitement a donc été ajoutée pour pallier cela. Il s'agit de l'assignation de l'orientation, qui se base sur la notion du moment d'une image. Pour une image numérique en niveau de gris, l'équation générale des moments est la suivante :

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{2.18}$$

A partir des moments, on peut déterminer le barycentre de l'objet dans l'image :

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \tag{2.19}$$

On peut construire un vecteur entre le centre du coin et le barycentre.L'orientation du patch est déterminée ensuite simplement par :

$$\theta = atan2(m_{01}, m_{10}) \tag{2.20}$$

Pour le descripteur du point (xi,yi) et de taille n, on définit la matrice 2xn suivante :

$$\begin{pmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_n \\ \mathbf{y}_1 & \dots & \mathbf{y}_n \end{pmatrix} \tag{2.21}$$

En utilisant l'orientation du patch, on peut obtenir la matrice de rotation qui lui correspond. Ainsi, on obtient une version orientée de la matrice S :

$$\mathbf{S}_\theta = \mathbf{R}_\theta \mathbf{S} \tag{2.22}$$

La caractéristique BRIEF devient maintenant :

$$\mathbf{g}_n(\mathbf{p}, \theta) := f_n(\mathbf{p}) | (x_i, y_i) \in \mathbf{S}_\theta \tag{2.23}$$

Afin de gagner en temps de calcul, un tableau de consultation est réalisé, reliant l'angle avec un pas de $\frac{2\pi}{30}$, à des modèles de descripteurs BRIEF.

2.4 SFOP

2.4.1 Introduction

SFOP(Scale invariant Feature OPerator) [18]est un opérateur de détection de points clés invariant à l'échelle. Il a été développé par FÖrstner à l'université de Bonn en 2009. Il permet une détection et une classification des structures significatives présentes sur l'image. Il se base sur le détecteur proposé par le même auteur en 1994, il ne s'agissait à l'époque que d'un détecteur de jonction. Il a été généralisé pour extraire les différents types de structures significatives présentes sur l'image. Il se base aussi sur le modèle général des fonctions spirales de Bigün (1990). SFOP ne détecte que les points clés. Il exploite la robustesse du SIFT pour le calcul des descripteurs.

2.4.2 Idée générale

La procédure suivie ici est contrôlée par des paramètres sémantiquement clairs. On obtient ainsi des points clés caractérisés par la position, l'échelle, le type et la mesure de la cohérence.
L'idée est de sélectionner des points où la cohérence est optimale, par rapport au modèle de la spirale.
Il est adapté spécialement aux structures de type cercle ou étoile.

2.4.3 Contexte

FÖrstner estime que ce détecteur permet de traiter les images pauvres en texture. Prenons par exemple l'image ci-dessous. Les détecteurs existant dans la littérature n'y détecteraient pas assez de points clés pour la caractériser. Ou alors, ils trouveraient des points clés ayant des positions très proches. Ceux-là devraient d'ailleurs être éliminés durant l'étape de mise en correspondance. On remarque donc que Harris ne détecte pas les coins présents sur l'image, comme on pourrait le croire. On remarque aussi que SFOP détecte plus de points que le SIFT.

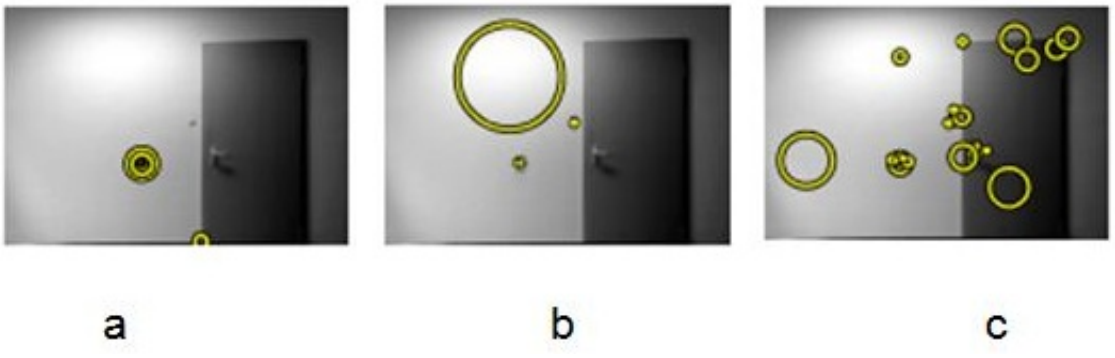

FIGURE 2.17: Détection de points clés par a) Harris b) SIFT c) SFOP

2.4.4 Propriétés essentielles des points clés

Selon FÖrstner, les points clés caractérisant une image devraient avoir les propriétés suivantes :

- **Complétion et complémentarité**
  Le détecteur doit détecter un maximum de structures signifiantes présentes sur l'image. Ainsi, la description de la scène sera complète. Cela veut dire aussi que des points complémentaires sont extraits de l'image.

- **Invariance et répétabilité**
  Les points clés doivent être invariants à l'échelle et à la rotation. Ils doivent fournir une bonne répétabilité afin de donner de bons résultats lors de la mise en correspondance des images.

- **Précision**
  Les points clés doivent être localisés avec une bonne précision sur l'image.

- **Interprétabilité**
  Les éléments basiques présents sur la scène, en particulier les cercles et les étoiles, doivent faire partie des points sélectionnés.

- **Paramètres de contrôle**
  Tous les points clés doivent avoir un ensemble de paramètres les caractérisant. Ces paramètres doivent être sémantiquement aussi clairs que possible.

2.4.5 le modèle spirale

Prenons un patch autour du pixel central P. La structure de ce patch sera considérée comme spirale idéale si la direction du gradient à un point q du voisinage, fait un angle α constant avec le vecteur radial. Cela est illustré dans la figure ci-dessous.

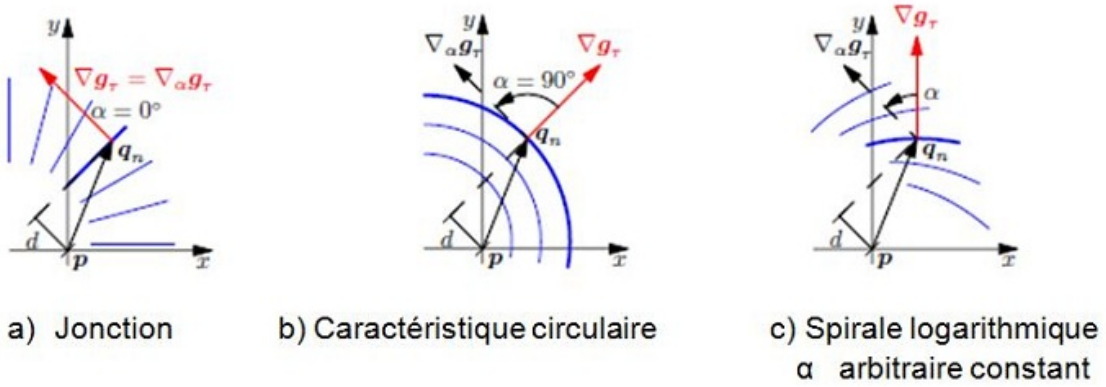

FIGURE 2.18: Mesure de l'angle αet de la distance d

Dans le cas $\alpha = 0°$, les directions des bords pointent vers le centre de rotation et forment une étoile. C'est le premier cas particulier (Figure 2.18 a). Le deuxième est pour $\alpha = 90°$. Les directions des gradients pointent vers le centre de rotation. On est en présence d'un cercle (Figure 2.18 b). Le troisième cas représente les structures ayant un α arbitraire. Ce sont des spirales logarithmiques. La section suivante explique l'algorithme qui va sélectionner les structures idéales.

2.4.6 L'algorithme

2.4.6.1 Construction de l'espace échelle

Comme pour les méthodes étudiées auparavant, cet algorithme construit un espace échelle pour la détection des points d'intérêts éventuels. Pour toutes les valeurs σ du filtre Gaussien appliqué, un tenseur de structure est calculé pour chaque pixel.

Tenseur de structure

Un tenseur de structure est la matrice représentant les directions du gradient d'une fonction. Il permet dans le cas d'une image de décrire les variations locales des structures présentes dessus. Il possède deux paramètres, une échelle d'intégration σ et une échelle

de différenciation τ. Le tenseur M d'une image U est une matrice 2×2 définie comme suit :

$$M = \begin{bmatrix} G_\tau * (I_x^2) & G_\tau * (I_x I_y) \\ G_\tau * (I_x I_y) & G_\tau * (I_y^2) \end{bmatrix} \tag{2.24}$$

Avec : $I_x = \frac{dU_\sigma}{dx}$ et $I_y = \frac{dU_\sigma}{dy}$ où : $U_\sigma = G_\sigma * U$

2.4.6.2 Calcul du poids de chaque pixel

La troisième étape consiste à calculer le poids de chaque pixel. Pour cela, la distance d_n entre la direction du gradient et le point P à travers tous les pixels q_n du voisinage de P est calculée.
Le gradient $\nabla_{Tg} = \nabla G_T * g$ dépend de l'échelle de différenciation τ.
En transformant la distance avec la matrice de rotation R_α, on obtient :

$$d(p, q_n, \alpha, T) = (q_n - p)^T \; R_\alpha \nabla_{Tg}(q_n) / |\nabla_{Tg}(q_n)| \tag{2.25}$$

La variance du point central est déterminée par la borne de Cramer Rao dérivée de l'estimation du maximum de vraisemblance de tous les pixels du voisinage du pixel P. La variable stochastique $d(q/M)$ de la distance d est supposée normalement distribuée. Elle aurait la moyenne nulle et la variance suivante :

$$var(d(q/M)) = \frac{s^2}{|\nabla_{Tg}(q_n)|^2 G_\sigma(q - p)} \tag{2.26}$$

Il est maintenant possible de calculer le poids. Il constitue tout simplement l'inverse de la variance de la distance :

$$w = \frac{1}{var(d(q/M))} \tag{2.27}$$

Le poids possède trois propriétés importantes :

- Il augmente avec le carré du module du gradient.
- Dépend de la distance du point q du point de référence p.
- s^2 représente son paramètre échelle (rapport de l'échelle d'intégration et de l'échelle de différenciation).

2.4.6.3 Détermination du poids optimal

Tout d'abord, les paramètres σ,τ,α sont fixés afin de pouvoir calculer la position optimale. La fonction de vraisemblance logarithmique négative suivante doit être minimale :

$$\Omega = \sum_{n=1}^{N(\sigma)} \left[(q_n - p)^T \; R_\alpha \nabla_\tau g(q_n)\right]^2 \;\; G_\sigma(q_n - p) \tag{2.28}$$

$N(\sigma)$ est le nombre de pixels du voisinage. Il a été fixé pratiquement à $N(\sigma) = 12\sigma^2 + 1$. On obtient ainsi la position $\hat{p}$ estimée.
On peut ensuite calculer la borne de Cramer Rao de la matrice de covariance de $\hat{p}$, qui est égale à :

$$\sum_{\hat{p}\hat{p}} = \widehat{s^2}\, M^{-1} \tag{2.29}$$

Avec

$$M = \sum_n J_n W J_n^T \tag{2.30}$$

Sachant que J_n représente la jacobienne de la distance : $J_n = \partial d_n / \partial p$ et W la diagonale du poids : $W = Diag(s^2/Var(\underline{d}_n\ ))$.
Le facteur de variance estimé est donné par :$\widehat{s^2} = \Omega(\hat{p})/(N\,(\sigma) - 2)$
Puis, afin d'obtenir une valeur scalaire pour la précision sur la position, on calcule la valeur propre maximale de la matrice de covariance $\lambda_1(\sum_{\hat{p}\hat{p}}\,)$. Celle-ci indique la variance maximale de la position estimée $\hat{p}$.
Pour trouver les points clés ayant la précision maximale sur la position, on prend l'inverse de la valeur propre maximale. On obtient l'expression de la précision suivante :

$$w\,(p, \alpha, \tau, \sigma) = \frac{1}{\lambda_1(\sum_{\hat{p}\hat{p}}\,)} = \frac{(N\,(\sigma) - 2)\,\lambda_2(M\,(p, \alpha, \tau, \sigma))}{\Omega\,(p, \alpha, \tau, \sigma)} \tag{2.31}$$

2.4.6.4 Estimation de l'angle alpha

Avant de sélectionner les points dont w est maximale, il est nécessaire de déterminer la valeur de α qui l'optimise. La somme $\Omega(\alpha)$ est périodique, et a pour expression :

$$\Omega\,(\alpha) = \alpha - b cos(2\alpha - 2\alpha_0) \tag{2.32}$$

Avec un minimum de $\Omega_{min} = a - b$ lorque $\alpha \;=\; \alpha_0$.
La détermination de a, b et σ_0 se fait en remplaçant α par les trois valeurs particulières

0°,60° et 120°. Le paramètre α_0 maximise la précision p pour une localisation et des échelles particulières.

2.4.6.5 Suppression des non-maxima

Une dernière étape permet de sélectionner les points d'intérêt définitifs. Elle comporte plusieurs critères de sélection :

1. Comparaison par rapport à la position : Pour celà,uniquement les points appartenant à la même catégorie de structures sont comparés par rapport à leurs positions.On calcule ainsi la distance de Mahalanobis entre deux points p et p' par exemple. Si cette distance est inférieure à un certain seuil, p' est éliminé.
   Mahalanobis
   En statistique, la distance de Mahalanobis est une mesure de distance basée sur la corrélation entre des variables. C'est une manière utile de déterminer la similarité entre une série de données. Elle diffère de la distance euclidienne par le fait qu'elle prenne en compte la corrélation de la série de données. Ainsi, à la différence de la distance euclidienne où toutes les composantes des vecteurs sont traitées de la même façon, la distance de Mahalanobis accorde un poids moins important aux composantes les plus bruitées.

2. A partir des octaves construites pour la détection des points clés, chaque pixel est entouré de 26 voisins : 9 pixel de l'octave du dessus et 9 pixel de l'octave du dessous, en plus des 8 pixels l'avoisinant sur son octave. Seul le pixel dont le poids est maximal est retenu.

3. Ce critère concerne la valeur propre minimale $\lambda_2(M)$ de la matrice tenseur de structure M. Un seuil $T_\lambda$est établi. Uniquement les points dont lambda est supérieur à ce seuil sont retenus.
   $$\lambda_2(M) > T_\lambda \tag{2.33}$$

Conclusion

Dans ce chapitre, nous avons découvert la théorie de chacun des algorithmes comparés, de manière détaillée.
Nous nous sommes aussi familiarisés avec le vocabulaire et les différents outils mathématiques qu'utilisent ces méthodes : ondelettes, image intégrale, filtres Gaussiens, octaves,

échelle,... etc.

Enfin, nous avons recueilli des informations concernant l'invariance de ces algorithmes par rapport aux variations de l'image.

Dans le chapitre suivant, nous exposons les résultats obtenus suite à la comparaison de ces algorithmes de manière pratique.

Chapitre 3

Tests et résultats

3.1 Introduction

Dans cette partie, nous présentons les résultats de comparaison des algorithmes sur lesquels nous avons travaillé : SIFT, SURF et ORB. Afin d'obtenir ces résultats, nous avons choisi plusieurs critères de comparaison. Ce sont les variations de l'image, par rapport auxquels les algorithmes étudiés sont dits invariants. Aussi, la comparaison a été faite en appliquant les algorithmes à notre base de données, contenant des images de modèles de voitures.

3.2 Base de données

La base de données que nous avons utilisée dans ce travail est un ensemble d'images de modèles de voitures. Elle est composée d'un dossier Training et d'un dossier Testing. Les deux dossiers contiennent respectivement 154 et 96 images de voitures, répertoriées selon la marque, le modèle et l'année de production comme détaillé au tableau 3.1 . Les images sont en format jpeg, en niveau de gris et de taille 150x66 pixels.

Marque	Modèle	Années de production	Nombre	
			Testing	Training
Audi	A4	98-04	4	7
BMW	Série 3	01-04	7	7
		95-98	3	7
Ford	Fiesta	03-06	6	7
		00-02	3	7
		96-99	6	7
	Focus	01-05	5	7
	Ka	97-05	5	7
	Mondeo	01-06	3	7
Peugeot	206	98-04	5	7
	306	97-01	10	7
Rover	400	93-99	3	7
Toyota	Avensis	97-01	3	7
	Corolla	00-03	3	7
	Yaris	01-05	4	7
Vauxhall	Astra	00-03	3	7
	Corsa	02-05	5	7
		95-00	3	7
Volkswagen	Golf	04-06	5	7
		02-03	3	7
	Passat	99-00	3	7
	Polo	95-99	4	7

TABLE 3.1: Details de la base de données

3.3 Application à la base de données

3.3.1 SIFT, SURF, SFOP et ORB sur Matlab

- Pour SIFT, nous avons exploité les codes écrits par Andrea Vedaldi, professeur à l'Université de Californie. Ces codes sont disponibles en téléchargement légal. Leur licence permet leur utilisation, modification et redistribution pour toute utilisation pédagogique ou pour la recherche à but non lucratif [19].

- Concerant SURF nous avons téléchargé la librairie OpenSurf, une librairie Open Source permettant l'utilisation de l'algorithme Surf à des fins pédagogiques non lucratives[20].

- Pour l'utilisation et l'évaluation de ORB, nous avons utilisé la librairie OpenCV, dans laquelle l'algorithme est implémenté. Nous avons téléchargé le kit MexOpenCV

qui permet de connecter cette librairie à Matlab et ainsi utiliser les descripteurs ORB[21].

- Pour SFOP, nous avons directement exploité les codes élaborés par l'auteur Forstner et son équipe, à l'université de Bonn, Allemagne. Ces codes permettent uniquement la détection des points d'intérêt. Le calcul des descripteurs ainsi que la mise en correspondance s'est faite à l'aide des codes SIFT, comme le propose Forstner[22].

Les codes de chacun des algorithmes ont été étudiés, exploités et adapté.
Les codes de SIFT et ORB permettent en plus du calcul des descripteurs, de mettre en correspondance les images.
Les codes du SURF permettent l'extraction des points clés et le calcul des descripteurs. Nous avons nous-mêmes écrit les programmes pour la mise en correspondance.
Le SFOP n'étant qu'un détecteur de points clés, les codes du SIFT pour le calcul des descripteurs et la mise en correspondance ont été utilisés.

3.3.2 Évaluation

Les étapes suivies lors de l'évaluation des algorithmes sont communes à tous les algorithmes et se présentent comme suit :

Étape 1

Nous avons calculé les descripteurs de toutes les images du dossier Training.

Étape 2

Pour chaque image du dossier Testing, les descripteurs sont calculés puis comparés à ceux calculés dans l'étape 1. Selon l'algorithme, différentes distances ont été utilisées :

- Distance Euclidienne pour le SIFT, SURF et le SFOP.
- Distance de Hamming pour ORB.
  Distance de Hamming

La distance de Hamming permet de comparer deux séquences logiques. Elle comptabilise le nombre de bits différents entre les deux séquences. Par exemple :

$$a = (011011) etb = (111001)$$

$$d = 1 + 0 + 0 + 0 + 1 + 0 = 2$$

Posons :
A :Ensemble des points clés de l'image test.
N_A :Nombre d'éléments de A.
B :Ensemble des points clés de l'image d'apprentissage.
N_B :Nombre d'éléments de B
$P_A \in A$ etD_p son descripteur.
La distance Euclidienne (ou Hamming) est calculée entre D_P et les N_B descripteurs de l'ensemble B. Afin de rechercher le plus proche voisin P_{B1} (séparé d'une distance $\mathcal{D}_1$) et le second plus proche voisin P_{B2} (séparé d'une distance $\mathcal{D}_2$).
Pour que la mise en correspondance $P_A \rightarrow P_A$ soit prise en considération, il faut que le critère suivant, appelé critère d'unicité, soit respecté :

$$D_1 * s < D_2 \tag{3.1}$$

Avec
s Seuil de valeur supérieure à 1, fixé au préalable.
Plusieurs tests ont été effectués afin de determiner le seuil s optimal pour chaque algorithme. La valeur $s = 1.8$ a été retenue pour le SURF et le SFOP, $s = 1.5$ pour l'ORB et $s = 1.7$ pour le SIFT.
Une autre mesure a été prise lors de la sélection des matchs corrects. Les mises en correspondance multiples avec un seul point clé ont été ôtées pour n'en retenir qu'une seule. Cela est effectué en ne gardant que le point ayant la distance euclidienne la plus petite.

Étape 3

Selon un ordre décroissant du nombre de points mis en correspondance, nous avons classé les images du Training. L'image classée en premier dans la liste représente la meilleure correspondance. La figure 3.1 illustre cela, en mettant en évidence chaque point de l'image test avec son correspondant de l'image d'apprentissage.

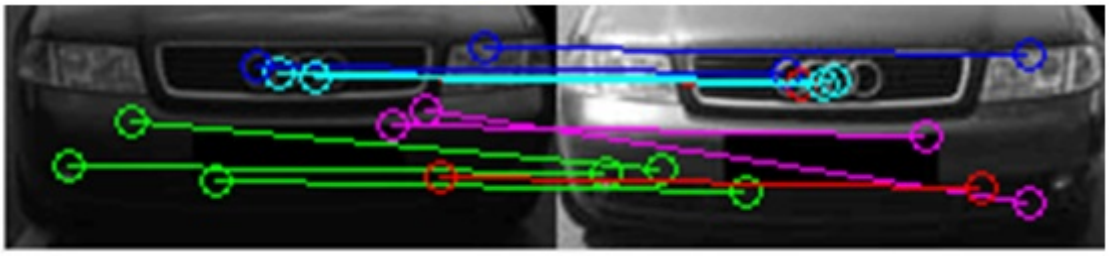

FIGURE 3.1: Image test (à gauche),image d'apprentissage (à droite)

La figure 3.2 illustre les étapes précédentes.

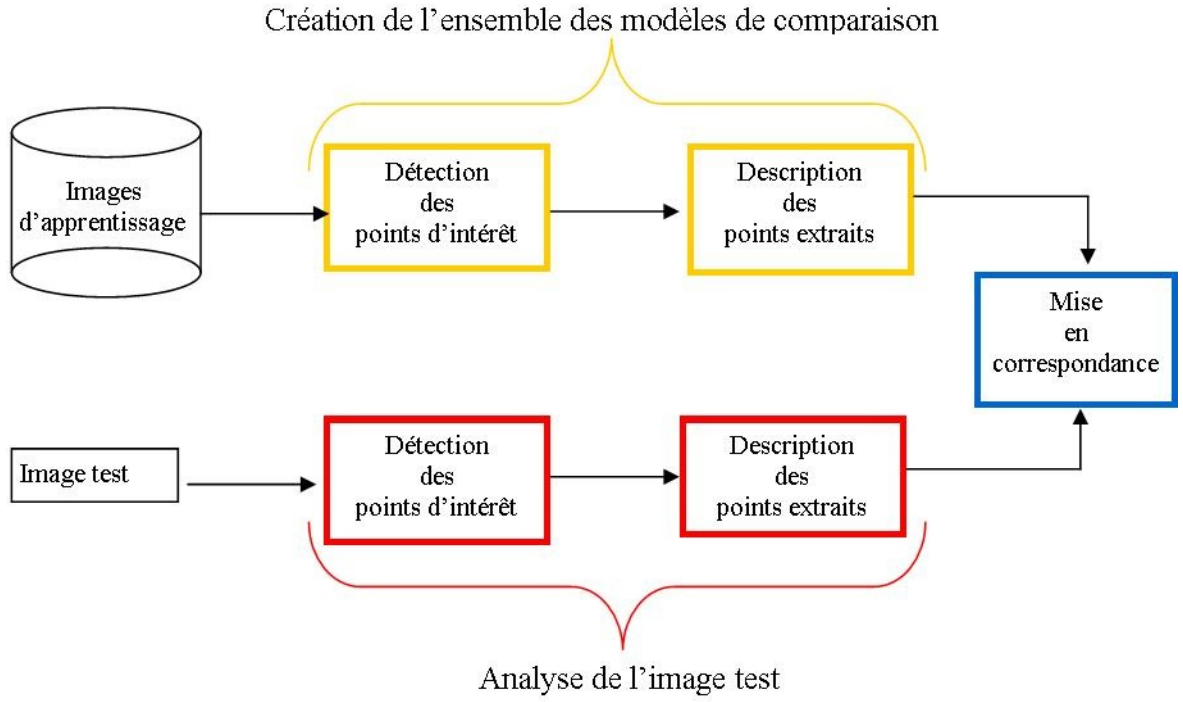

FIGURE 3.2: Algorithme de mise en correspondance

Étape 4

Dans la figure 3.1, le modèle de la voiture sur l'image test, correspond effectivement au modèle sur l'image d'apprentissage représentant la meilleure correspondance. Ce qui n'est pas toujours le cas comme dans la figure 3.3

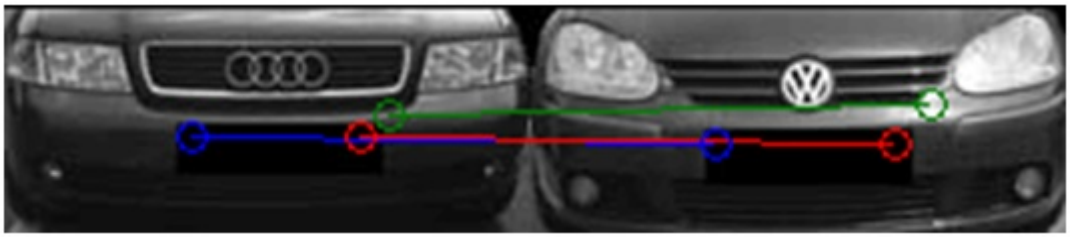

FIGURE 3.3: Fausse correspondance

- Afin d'évaluer le pourcentage de reconnaissance, nous avons écrit des programmes qui calculent le nombre de mises en correspondance correctes sur un ensemble de 96 images du dossier Testing. Nous avons appelé ce pourcentage le Top 1.

- A chaque modèle de voiture, dans le dossier Training ou Testing, correspond un chiffre. Si l'image d'apprentissage matchée avec l'image test possède le même chiffre, la mise en correspondance est considérée comme correcte. Dans la figure 3.1, le modèle présent sur les deux images correspond au chiffre 1. Alors que dans la figure 3.1, le modèle présent que l'image Test possède le chiffre 1 alors que celle de l'image d'apprentissage correspond au chiffre 20. C'est donc une correspondance incorrecte.

- Nous avons aussi calculé le pourcentage pour que l'image représentant la mise en correspondance correcte, soit parmi les 2 meilleures correspondances selon l'algorithme. Nous l'avons appelé le Top2.

- De plus, nous avons calculé le pourcentage pour que l'image représentant la mise en correspondance correcte apparaisse parmi les 5 meilleures correspondances de l'algorithme. Nous l'avons appelé le Top5.

- Les pourcentages du Top1, Top2 et Top5 ont aussi été calculés en variant plusieurs paramètres de l'image, afin d'élargir les critères de comparaison des algorithmes. Les critères sont les variations de luminosité, d'échelle, de rotation, de qualité de l'image, du bruit sur l'image, et de l'occultation de l'objet sur l'image.

3.3.3 Critères de comparaison

3.3.3.1 Luminosité

Afin de tester la robustesse des algorithmes étudiés vis-à-vis de la luminosité, nous avons éclairci et assombri les images de test à différents degrés. Cela nous a permis de calculer le pourcentage de reconnaissance à chaque fois.

Instruction Matlab utilisée

```
J=imadjust(I,[],[],gamma);
```

Avec

- I : Image originale.
- *gamma* : varie entre 0.2 et 1.8
  - Pour gamme=1, l'image est inchangée.
  - Plus gamma est inferieur à 1, plus l'image est éclaircie.
  - Plus gamma est superieur à 1, plus l'image est assombrie.
- J : Image assombrie ou éclaircie

FIGURE 3.4: Image originale

FIGURE 3.5: Image éclaircie

FIGURE 3.6: Image assombrie

3.3.3.2 Rotation

Le deuxième critère d'évaluation utilisé est la rotation. Nous avons donc testé l'efficacité des quatre méthodes en variant l'inclinaison de la voiture dans l'image entre -45° à +45° par pas de 10°.

Instruction Matlab utilisée

```
J=imrotate(I,angle,'bilinear','crop');
```

Avec

- I : Image originale.
- angle : Angle de rotation en degré.
- Options :
  - 'bilinear' : Pour que l'interpolation soit bilineaire.
  - 'crop' : Pour que l'image résultat.
- J : Image réorientée

FIGURE 3.7: Image originale

FIGURE 3.8: Rotation de 30°

3.3.3.3 Bruit

Quand l'objet à reconnaitre se trouve sur une image bruitée, l'efficacité de la reconnaissance est affectée. Pour étudier ce paramètre, nous avons ajouté aux images test un bruit Gaussien. En augmentant progressivement son écart-type (de 0.01 à 0.05 par pas de 0 .01), nous avons calculé le pourcentage de reconnaissance à chaque fois.

Instruction Matlab utilisée

```
J=imnoise(I,'gaussian',0,sigmabruit);
```

Avec

- I :Image originale.
- 'gaussian' : Type de bruit.
- 0 : Represente la moyenne.
- sigmabruit :la variance (nous l'avons variée de 0.01 à 0.05).
- J :Image bruitée

FIGURE 3.9: Image originale

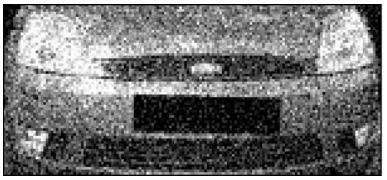

FIGURE 3.10: Image bruitée

3.3.3.4 Qualité

La qualité n'est pas toujours présente lors de la prise d'une image. Cela peut être dû à l'instabilité de la position de l'appareil. Nous avons ainsi simulé cette situation en appliquant un filtre Gaussien aux images test. L'écart- type du filtre Gaussien a été varié de 0.5 à 5 avec un pas de 0.5. En augmentant l'écart-type, l'image est d'autant plus floue.

Code Matlab utilisé

Cette opération s'effectue en deux temps :

- Définir un filtre Gaussien

  ```
  h=fspecial('gaussian',7,sigmaflou);
  ```

  Avec

  - 'gausian' : Type du filtre.
  - 7 : Taille du masque utilisé.
  - sigmaflou : Variance du filtre Gaussien.
  - h : Filtre résultant.

- Apliquer le filtre à l'image

  ```
  J=imfilter(I,h);
  ```

  Avec

  - I : Image originale.
  - h : Filtre défini par l'instruction fspecial.
  - J : Image floue.

FIGURE 3.11: Image originale

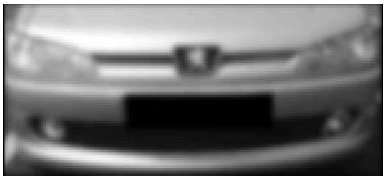

FIGURE 3.12: Image floue

3.3.3.5 Echelle

Les objets à reconnaitre ne sont parfois pas à la même taille sur leur image et dans les images des références. Nous avons donc agrandi et rétréci les images pour tester l'invariance des algorithmes par rapport à l'échelle. Nous avons fait varier celle-ci dans l'intervalle [0.75 ; 3] par pas de 0.25.

Instruction Matlab utilisée

```
J=imresize(I,echelle);
```

Avec

- I : Image originale.
- echelle : rapport entre dimensions de l'image résultat et image originale
- J : Image redimensionnée.

FIGURE 3.13: Image originale

FIGURE 3.14: Image redimensionnée

3.3.3.6 Occultation

Souvent, pour des acquisitions dans des conditions réelles, les objets à reconnaitre sont partiellement occultés. Nous avons donc tenté de simuler cette occultation en cachant (forçant les pixels à zéro) un pourcentage du centre de l'image. Il n'existe évidement pas d'instruction Matlab permettant cette opération, nous avons donc écrit une fonction pour la réaliser .

Code Matlab utilisé

```
function J=occlusion(I,pourcent)
   [h,l]=size(I); %    Taille de l'image originale.
   J=I;    %   Image sortie=image entrée.
   e=floor(pourcent*l/100); %  Calcul du nombre de colonne à cacher.
   lm=floor(l/2);  %   Colonne centrale.
   em=floor(e/2);  %   Motié du nombre de colonne a cacher.
   J(:,lm-em+1:lm-em+e)=uint8(zeros(h,e));% Mise à zéro des pixels
                                          % des colonnes au centre.
end
```

Avec

- I : Image originale.
- pourcent : Pourcentage de l'image à occulter.
- J : Image de sortie.

FIGURE 3.15: Image originale

FIGURE 3.16: Occultation verticale centrée de 35% de l'image

3.3.4 Interface pour utilisateurs

Afin de faciliter l'utilisation et l'évaluation des algorithmes étudiés précédemment, nous avons créé une interface graphique sur Matlab. L'interface est dotée d'un menu déroulant afin de choisir l'image test. Elle contient aussi plusieurs boutons sliders permettant de modifier les différents paramètres de l'image comme la luminosité, l'échelle etc. Ces boutons sliders permettent de tester l'invariance des algorithmes quant aux critères cités plus haut.La figure 3.17 illustre cette interface.

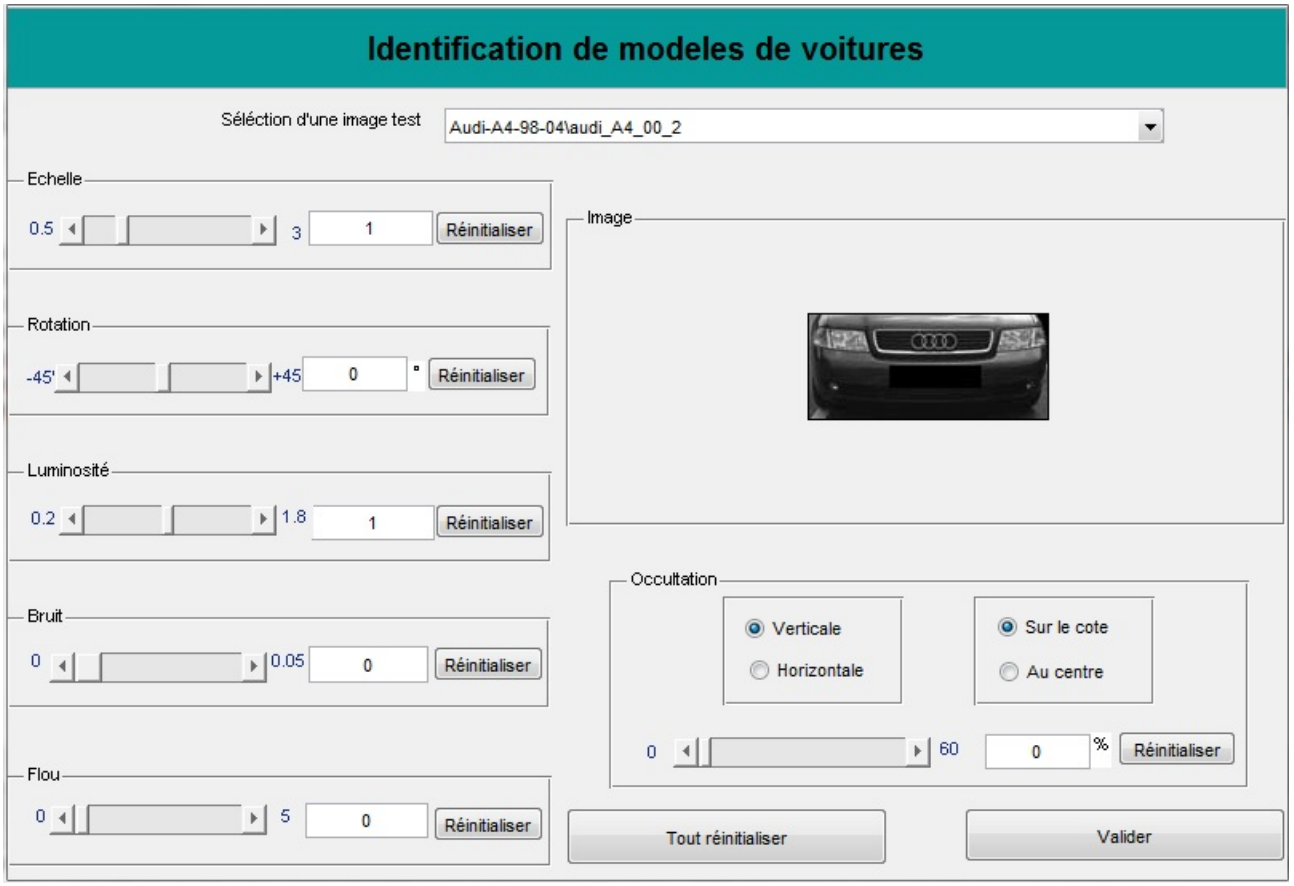

FIGURE 3.17: Interface graphique

Une fois l'image test choisie et les différents paramètres fixés, le bouton valider permet d'aboutir à de nouvelles fonctionnalités de l'interface. On peut en outre,au moyen de boutons radio, afficher la meilleures correspondances obtenues par chaque algorithme (figure 3.18), ou afficher le Top5 de chacun des algorithmes individuellement (figure 3.19)

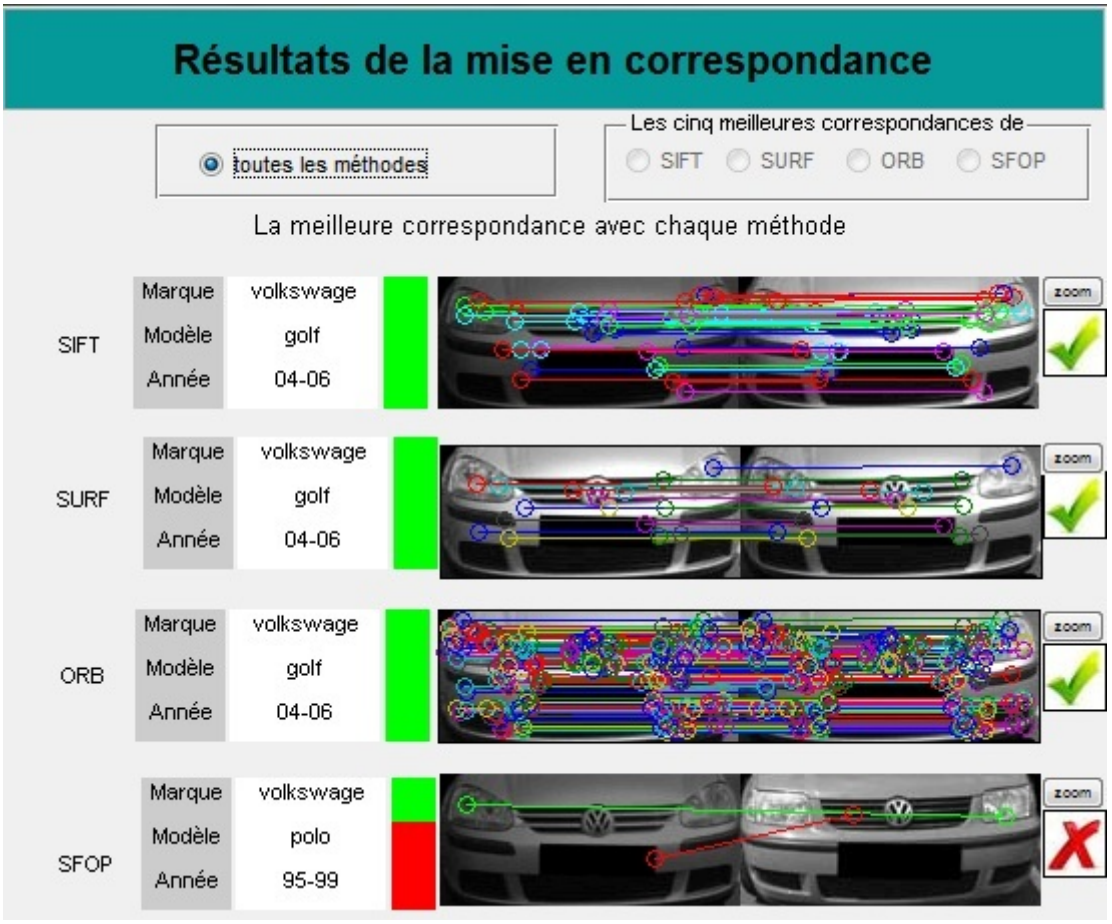

FIGURE 3.18: Meilleures correspondances avec chaque algorithme

Comme on peut le voir sur les figures 3.17,3.18 et 3.19, l'interface permet non seulement d'afficher les résultats des correspondances des algorithmes, mais aussi de les évaluer. Le check vert signifie que la mise en correspondance est correcte. La croix rouge signifie qu'elle est incorrecte. L'évaluation concerne aussi la marque, ou le modèle. Par exemple, si l'image d'apprentissage mise en correspondance est de la même marque que la voiture sur l'image test, un rectangle vert s'affiche à coté. Un rectangle rouge s'affiche dans le cas contraire.

Un bouton 'zoom' existe à coté de chaque mise en correspondance. Il permet de mieux voir les points d'intérêt détectés, éliminés ou matchés.Comme le montre la figure 3.20

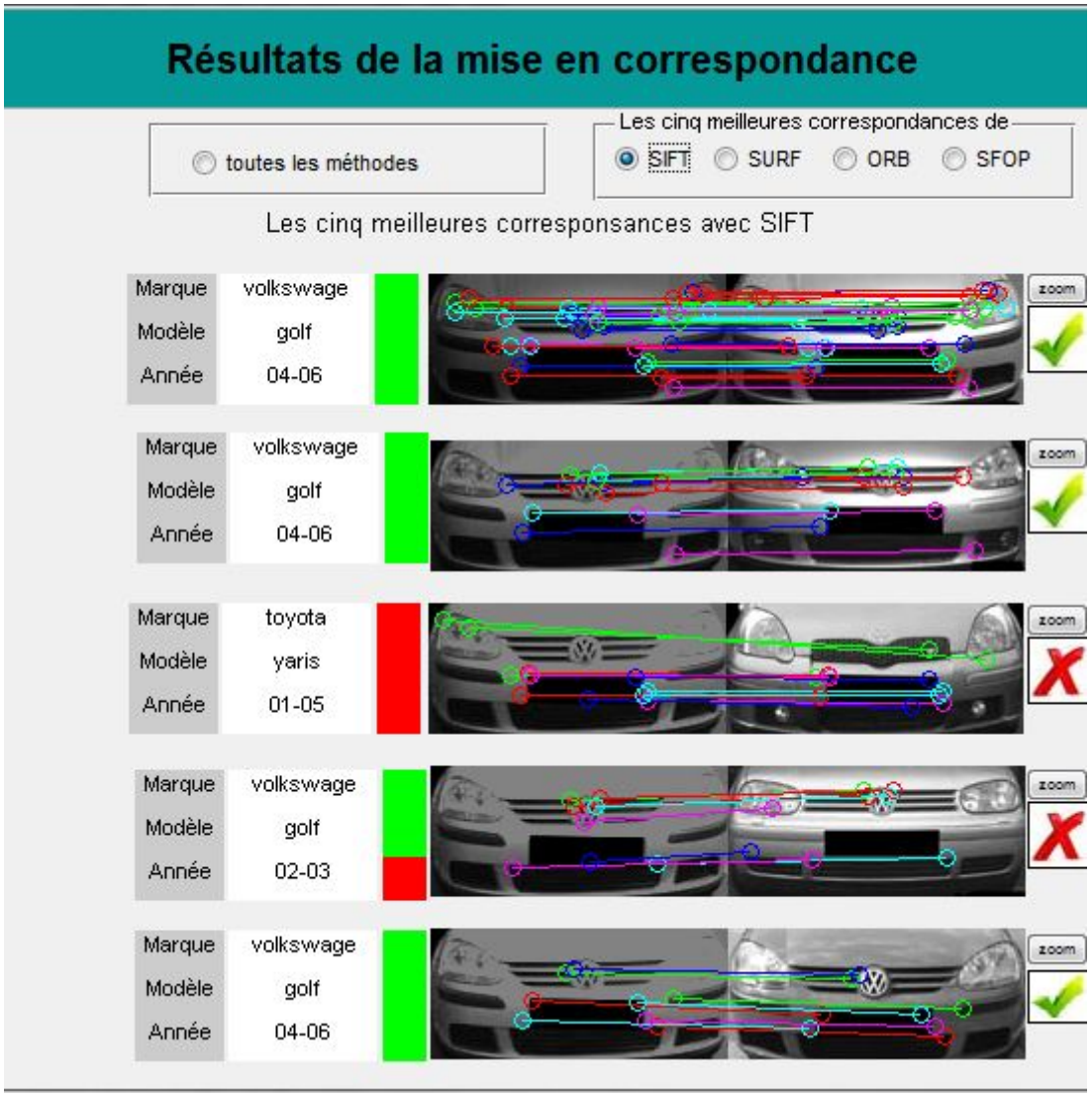

FIGURE 3.19: Les 5 meilleures correspondances selon le SIFT

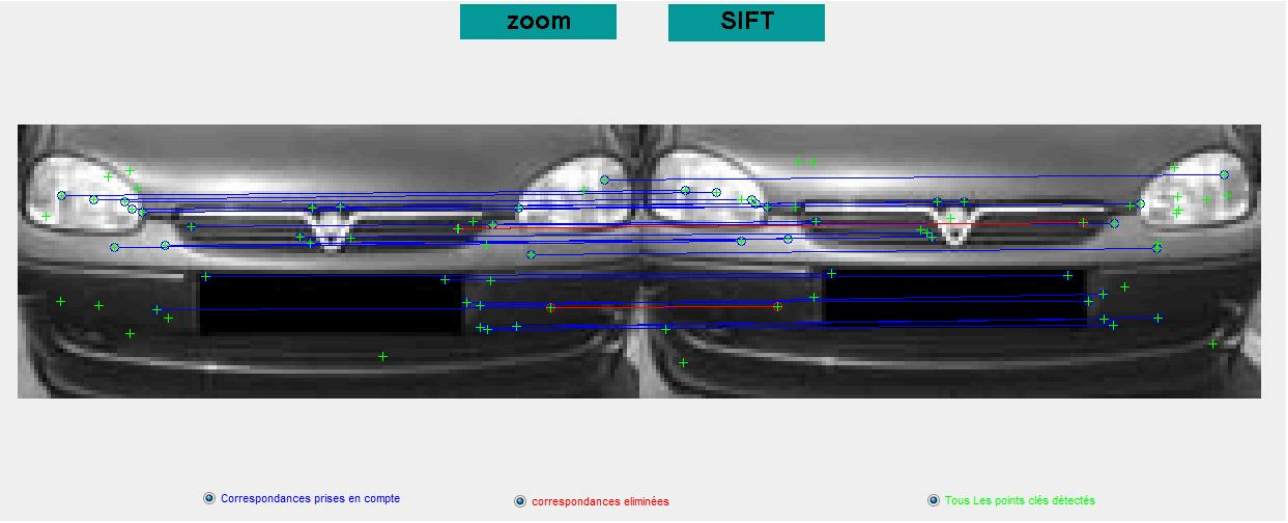

FIGURE 3.20: Zoom dans le cas SIFT

3.4 Evaluation

Nous avons tous d'abord calculé les pourcentages de reconnaissance sans aucun changement effectué sur l'image. Nous avons obtenu le tableau suivant :

	SIFT	ORB	SURF	SFOP
Top1	91.667	88.541	78.125	81.250
Top2	96.875	92.708	86.458	84.375
Top5	98.958	96.875	92.708	93.750

TABLE 3.2: Poucentages de reconnaissance sans aucun changement

Puis, en modifiant les différents critères cités plus haut nous avons obtenu ceci :

3.4.1 TOP 1

3.4.1.1 En variant la luminosité de l'image

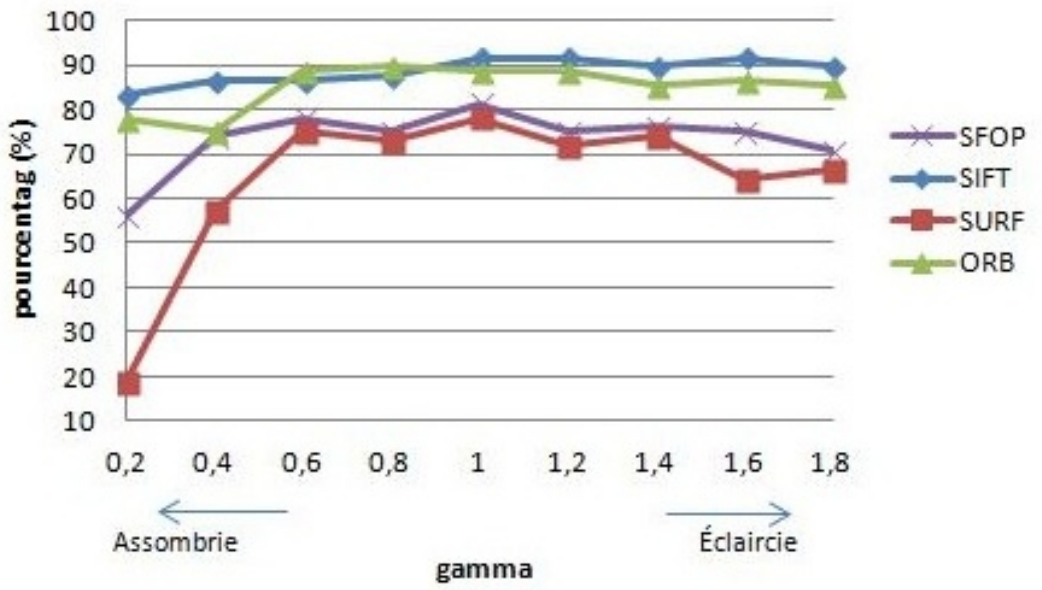

FIGURE 3.21: Reconnaissance du modèle à travers la meilleure correspondance en fonction de la luminosité

3.4.1.2 En variant l'angle de rotation de l'image

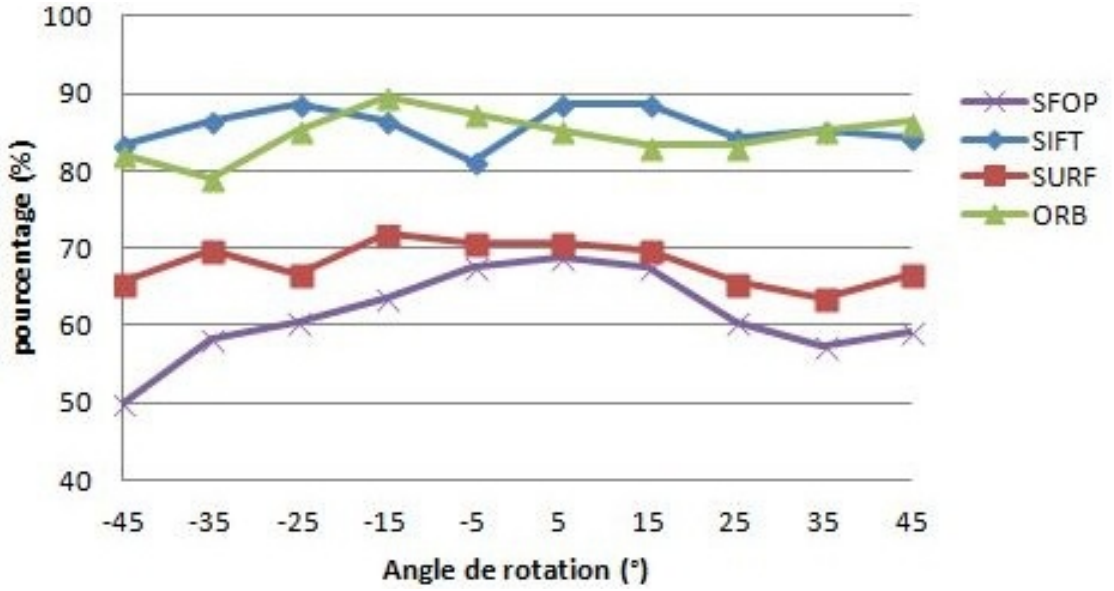

FIGURE 3.22: Reconnaissance du modèle à travers la meilleure correspondance en fonction de la rotation

3.4.1.3 Sensibilité au bruit

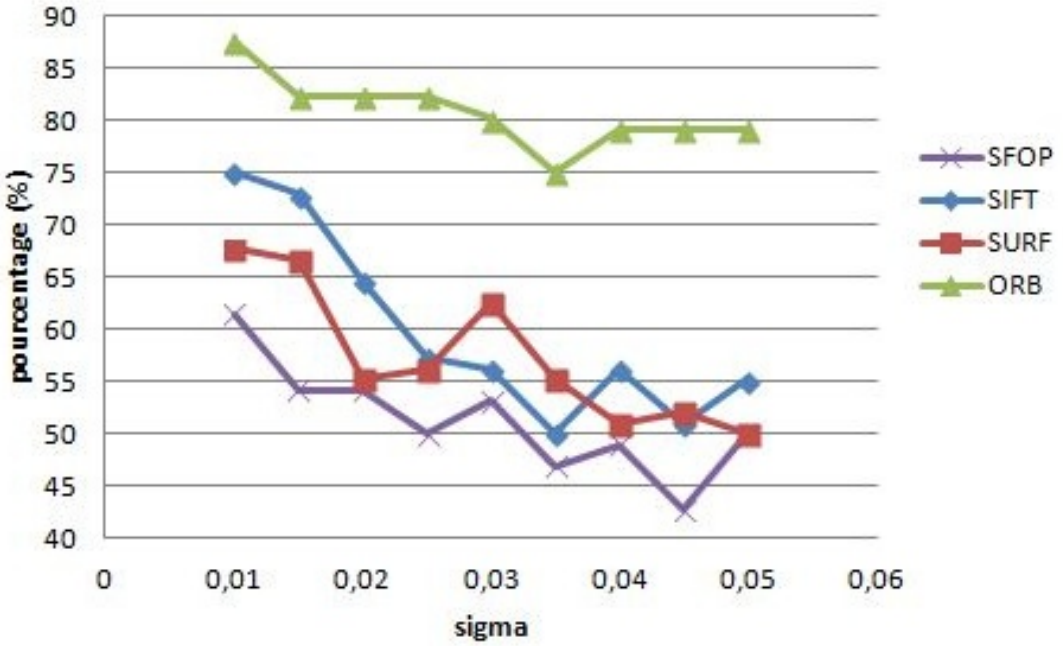

FIGURE 3.23: Reconnaissance du modèle à travers la meilleure correspondance en fonction du bruit

3.4.1.4 En variant la qualité de l'image

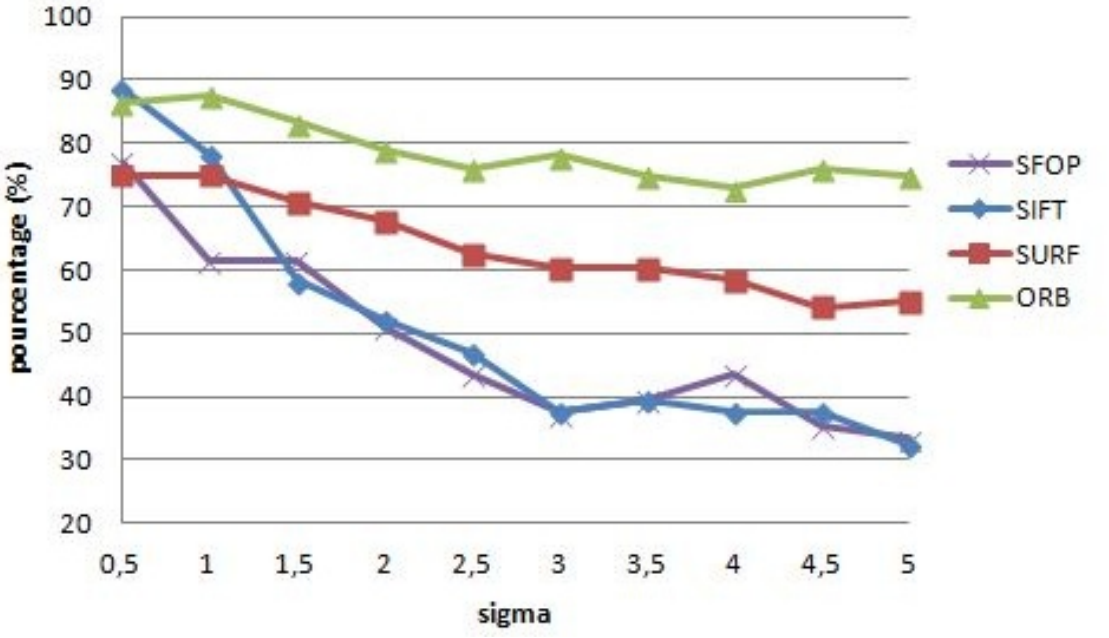

FIGURE 3.24: Reconnaissance du modèle à travers la meilleure correspondance en fonction de la qualité

3.4.1.5 Sensibilité aux occultations (Occultation verticale centrée)

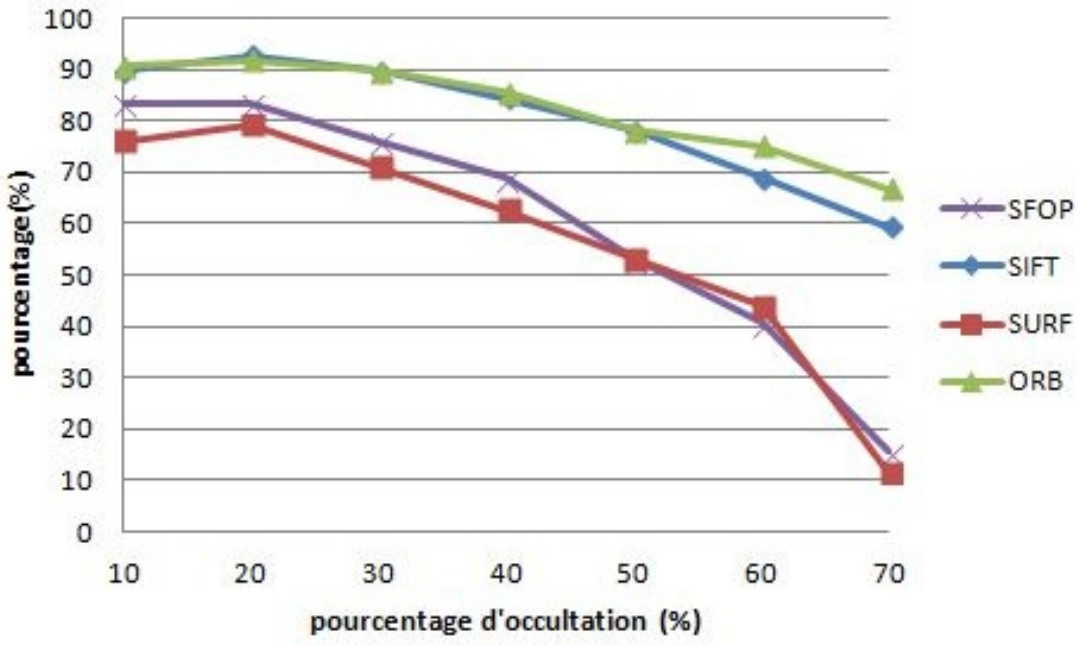

FIGURE 3.25: Reconnaissance du modèle à travers la meilleure correspondance en fonction des occultations

3.4.1.6 En variant l'échelle de l'image

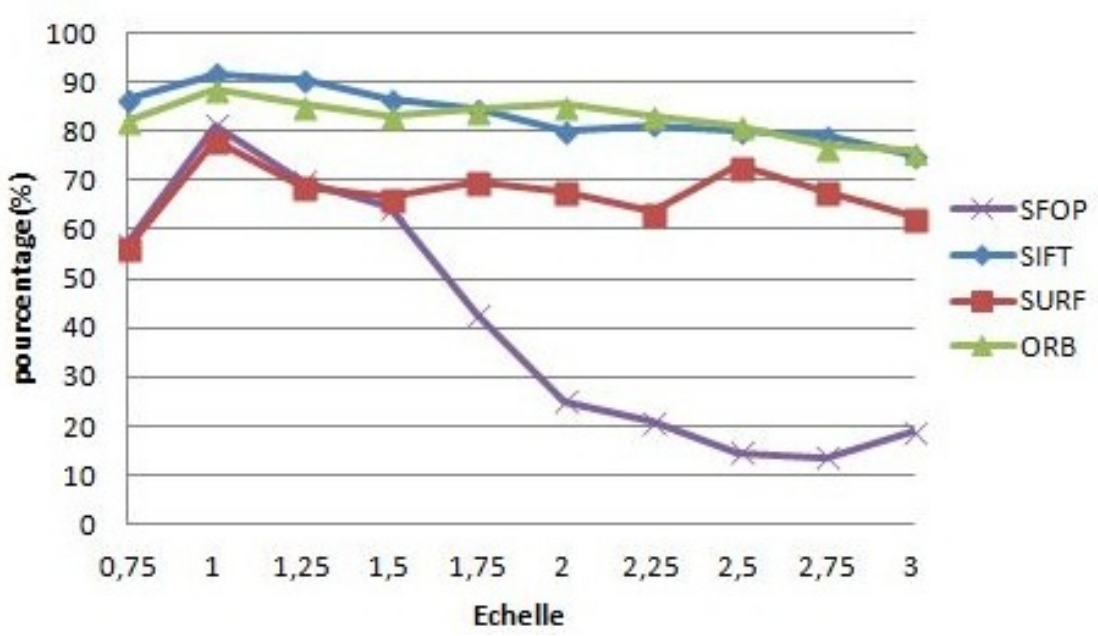

FIGURE 3.26: Reconnaissance du modèle à travers la meilleure correspondance en fonction de l'échelle

3.4.2 Tableau récapitulatif

Méthodes \critères		Luminosité	Rotation	Bruit	Flou	Echelle	Occultation
SIFT	% minimum	83,333	81,250	50,000	32,292	75,000	55,208
	% maximum	91,667	88,542	75,000	88,542	91,667	85,417
	% moyen	88,657	85,729	59,838	50,833	83,542	74,405
SURF	min	18,750	63,542	50,000	54,167	56,250	5,208
	max	78,125	71,875	67,708	75,000	78,125	71,875
	moy	64,352	68,125	57,407	63,958	67,396	48,363
ORB	min	75,000	79,167	75,000	72,917	76,042	58,333
	max	89,583	89,583	87,500	87,500	88,542	88,542
	moy	85,069	84,792	80,787	78,958	82,708	76,339
SFOP	min	56,250	50,000	42,708	33,333	13,542	6,250
	max	81,250	68,750	61,458	77,083	81,250	76,042
	moy	73,495	61,354	51,273	48,438	40,833	50,893

Légende: meilleur | second | troisième | dernier

TABLE 3.3: Resumé Top1

3.4.3 TOP 2

3.4.3.1 En variant la luminosité de l'image

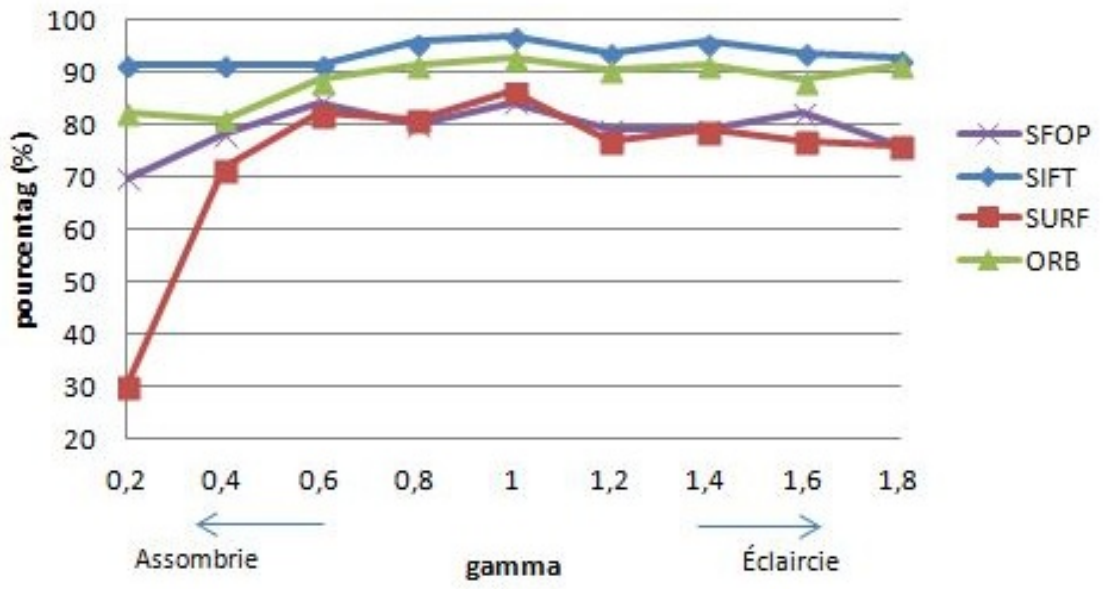

FIGURE 3.27: Reconnaissance du modèle à travers les deux meilleures correspondances en fonction de la luminosité

3.4.3.2 En variant l'angle de rotation de l'image

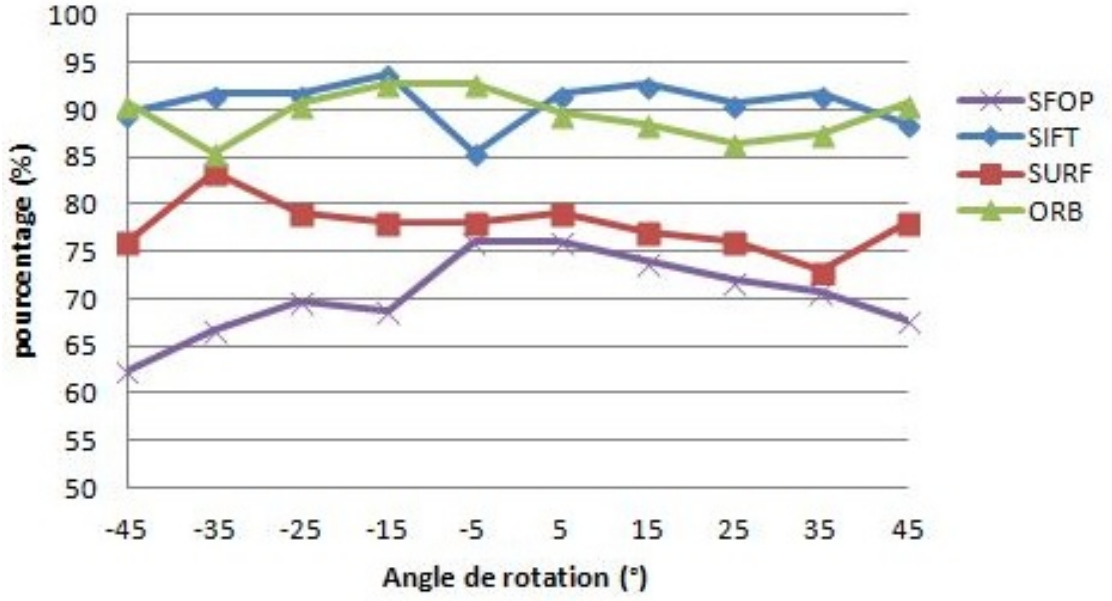

FIGURE 3.28: Reconnaissance du modèle à travers les deux meilleures correspondances en fonction de la rotation

3.4.3.3 Sensibilité au bruit

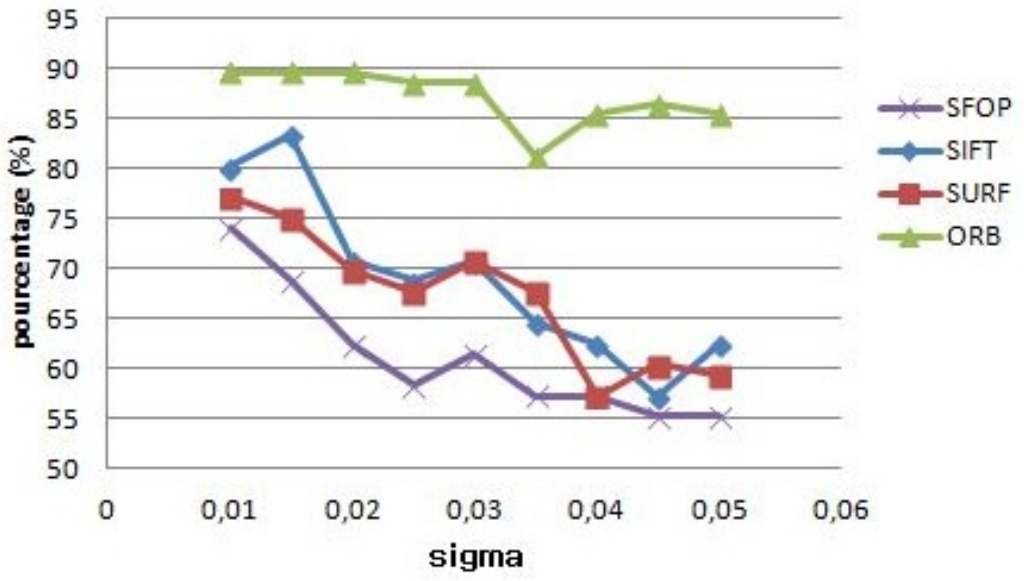

FIGURE 3.29: Reconnaissance du modèle à travers les deux meilleures correspondances en fonction du bruit

3.4.3.4 En variant la qualité de l'image

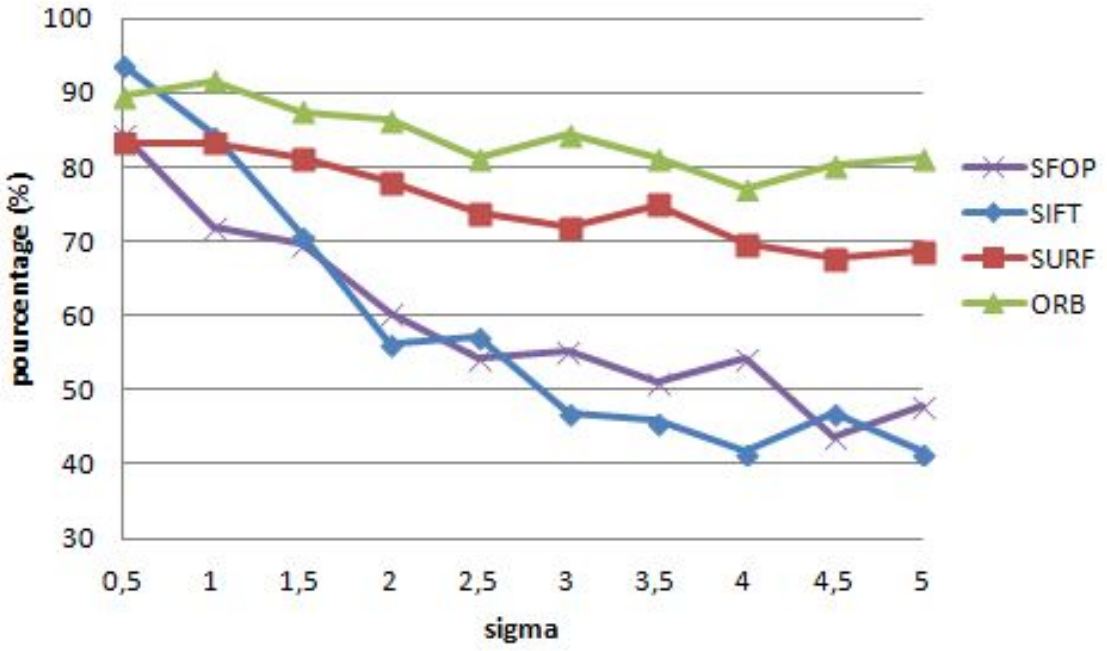

FIGURE 3.30: Reconnaissance du modèle à travers les deux meilleures correspondances en fonction de la qualité

3.4.3.5 Sensibilité aux occultations (Occultation verticale centrée)

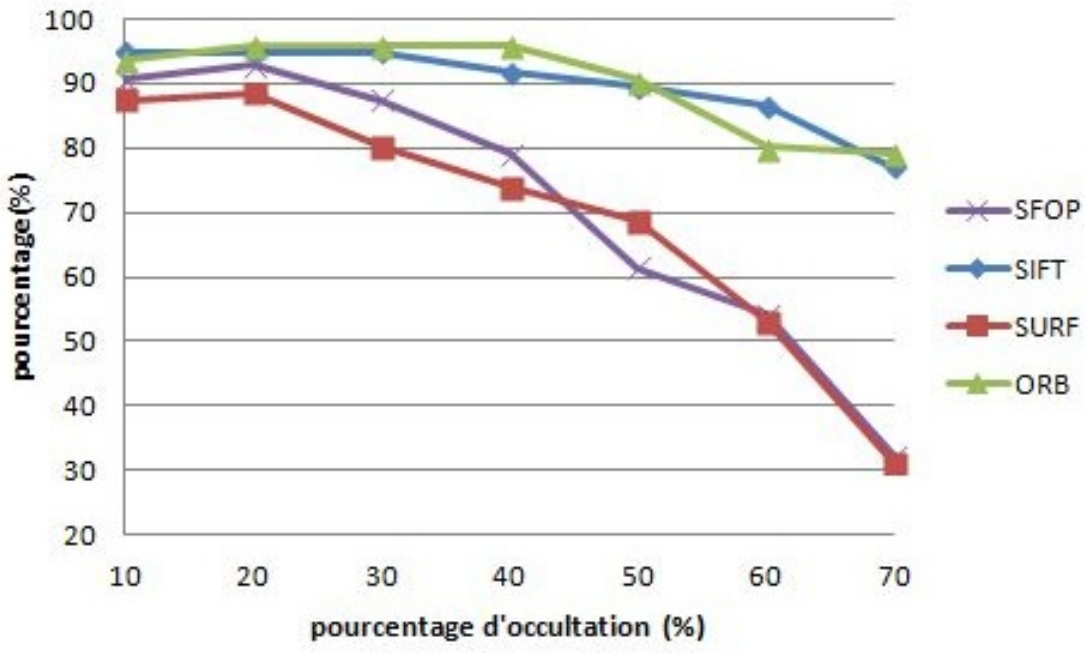

FIGURE 3.31: Reconnaissance du modèle à travers les deux meilleures correspondances en fonction des occultations

3.4.3.6 En variant l'échelle de l'image

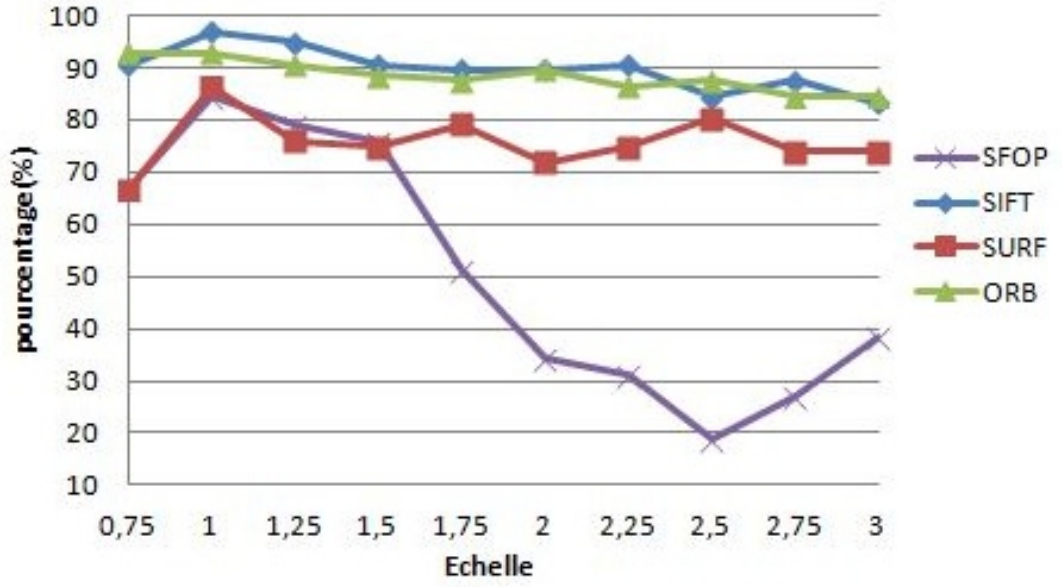

FIGURE 3.32: Reconnaissance du modèle à travers les deux meilleures correspondances en fonction de l'échelle

3.4.4 Tableau récapitulatif

Méthodes \critères		Luminosité	Rotation	Bruit	Flou	Echelle	Occultation
SIFT	% minimum	91,667	85,417	57,292	41,667	83,333	59,375
	% maximum	96,875	93,750	83,333	93,750	96,875	92,708
	% moyen	93,750	90,729	68,981	58,542	89,792	80,357
SURF	min	30,208	72,917	57,292	67,708	66,667	11,458
	max	86,458	83,333	77,083	83,333	86,458	79,167
	moy	73,495	77,813	67,245	75,313	75,833	56,696
ORB	min	81,250	85,417	81,250	77,083	84,375	66,667
	max	92,708	92,708	89,583	91,667	92,708	91,667
	moy	88,773	89,479	87,153	84,063	88,438	82,440
SFOP	min	69,792	62,500	55,208	43,750	18,750	15,625
	max	84,375	76,042	73,958	84,375	84,375	83,333
	moy	79,282	70,417	61,111	59,271	50,729	60,119

Légende: meilleur | second | troisième | dernier

TABLE 3.4: Resumé Top2

3.4.5 TOP 5

3.4.5.1 En variant la luminosité de l'image

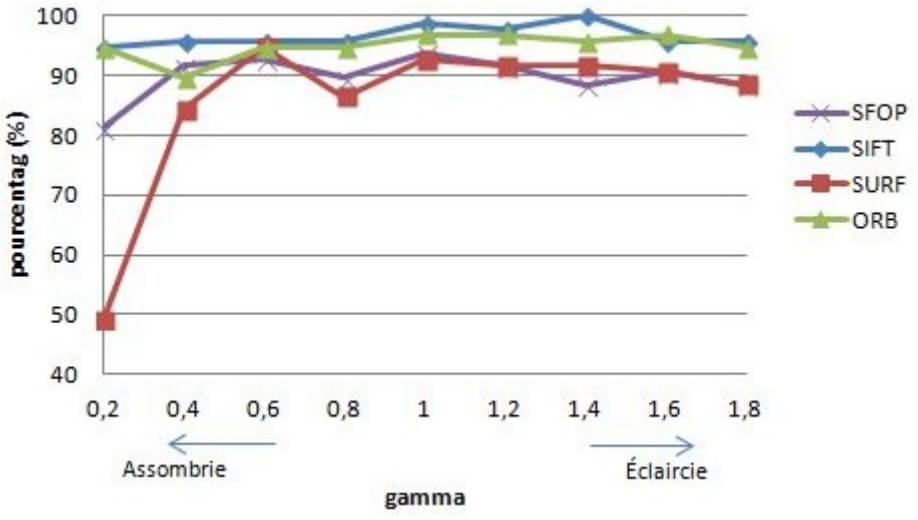

FIGURE 3.33: Reconnaissance du modèle à travers les cinq meilleures correspondances en fonction de la luminosité

3.4.5.2 En variant l'angle de rotation de l'image

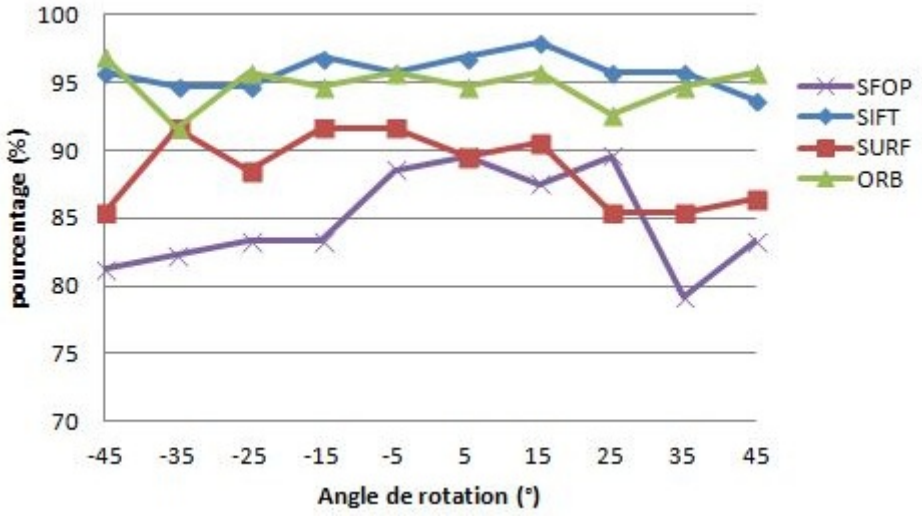

FIGURE 3.34: Reconnaissance du modèle à travers les cinq meilleures correspondances en fonction de la rotation

3.4.5.3 Sensibilité au bruit

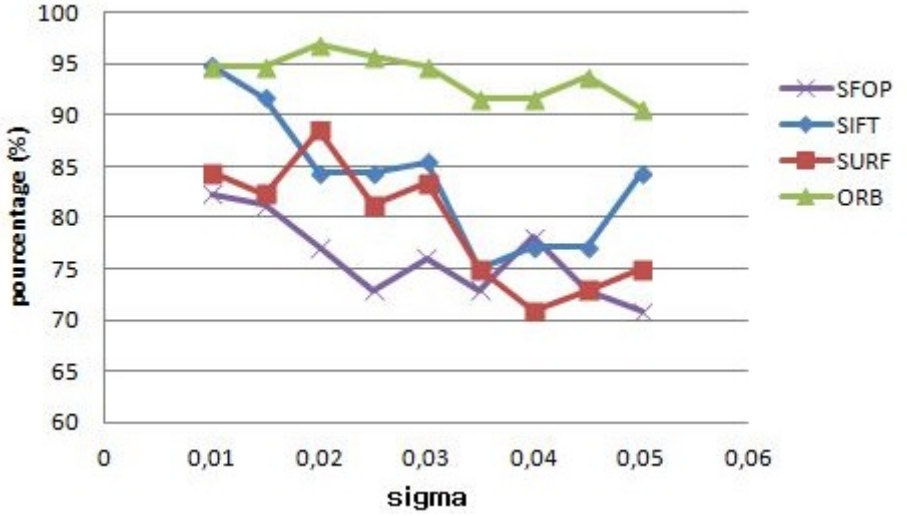

FIGURE 3.35: Reconnaissance du modèle à travers les cinq meilleures correspondances en fonction du bruit

3.4.5.4 En variant la qualité de l'image

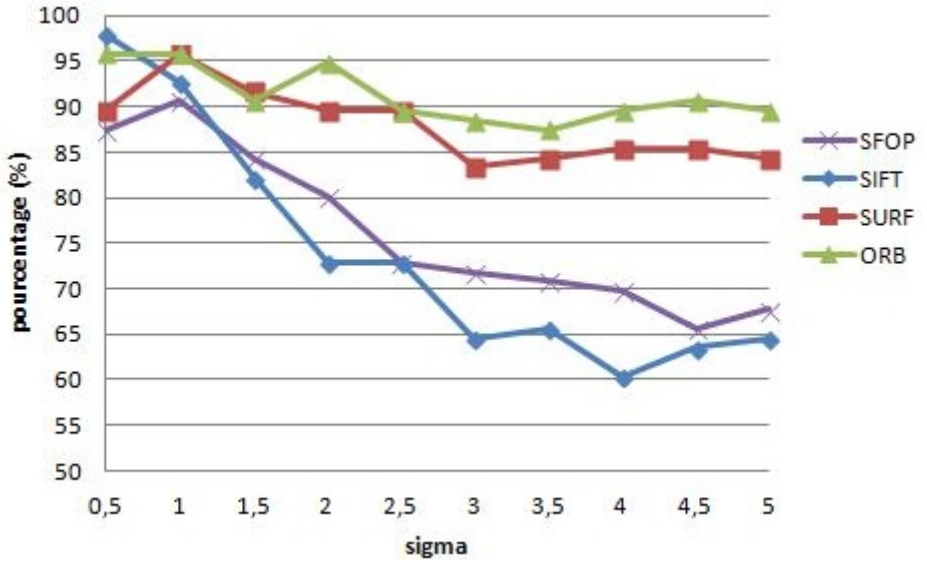

FIGURE 3.36: Reconnaissance du modèle à travers les cinq meilleures correspondances en fonction de la qualité

3.4.5.5 Sensibilité aux occultations (Occultation verticale centrée)

=

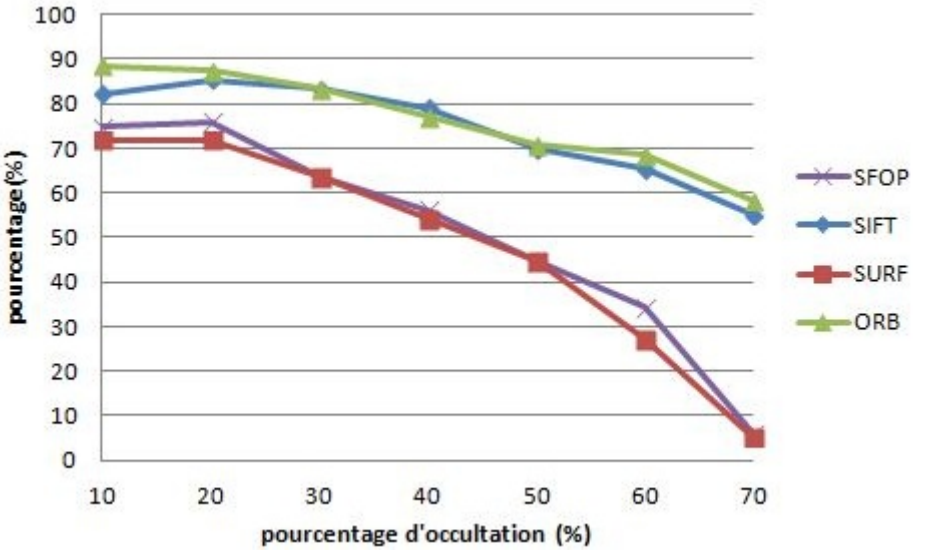

FIGURE 3.37: Reconnaissance du modèle à travers les cinq meilleures correspondances en fonction des occultations

3.4.5.6 En variant l'échelle de l'image

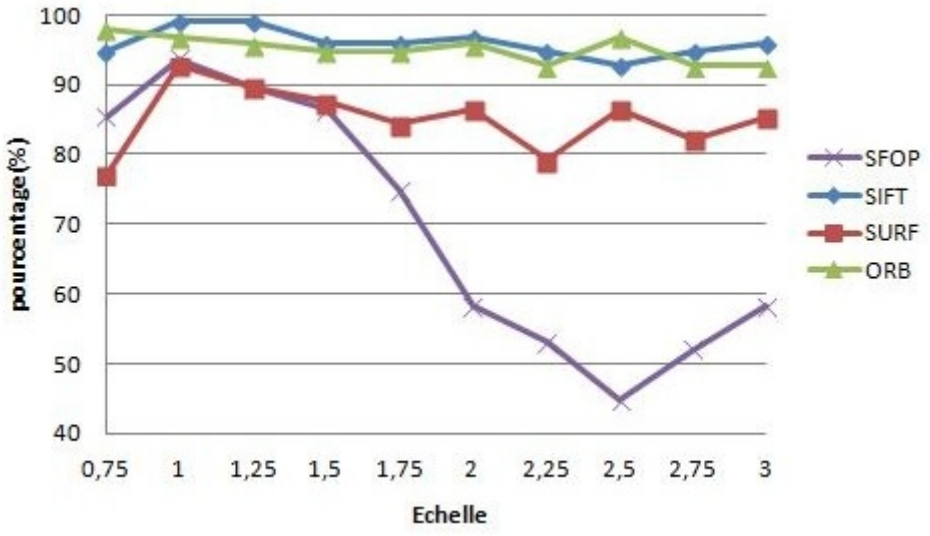

FIGURE 3.38: Reconnaissance du modèle à travers les cinq meilleures correspondances en fonction de l'échelle

3.4.6 Tableau récapitulatif

Méthodes \critères		Luminosité	Rotation	Bruit	Flou	Echelle	Occultation
SIFT	% minimum	94,792	93,750	75,000	60,417	92,708	77,083
	% maximum	100,000	97,917	94,792	97,917	98,958	94,792
	% moyen	96,759	95,833	83,796	73,750	95,938	89,881
SURF	min	48,958	85,417	70,833	83,333	77,083	31,250
	max	94,792	91,667	88,542	95,833	92,708	88,542
	moy	85,532	88,646	79,282	87,917	85,104	69,048
ORB	min	89,583	91,667	90,625	87,500	92,708	79,167
	max	96,875	96,875	96,875	95,833	97,917	95,833
	moy	95,023	94,896	93,866	91,250	95,104	90,179
SFOP	min	81,250	79,167	70,833	65,625	44,792	32,292
	max	93,750	89,583	82,292	90,625	93,750	92,708
	moy	89,815	84,792	76,042	76,146	69,688	71,131

Légende	meilleur	second	troisième	dernier

TABLE 3.5: Resumé Top5

3.5 Interprétation des résultats

3.5.1 Luminosité

D'après les graphes obtenus pour le top1, le SIFT et l'ORB donnent des résultats quasi similaires, qui se situent entre 75 et 90%. Le SURF affiche de mauvais résultats et descend jusqu'à moins de 20%. Le SFOP donne des résultats légèrement meilleurs que ceux du SURF mais restent inférieurs à ceux du SIFT et de ORB.
Dans le top2 et le top5, le SIFT donne de meilleurs résultats que les autres en atteignant un 100%. Néanmoins, ORB affiche de bons résultats aussi, entre 81 et 97%. Les statistiques du SURF et du SFOP restent largement inférieures à celles du SIFT et de ORB, et atteignent rarement les 90%. Ils affichent d'ailleurs des valeurs similaires sauf pour la première valeur de Gamma, où le SURF donne une valeur très basse par rapport à la moyenne.

3.5.2 Rotation

Pour la rotation aussi, le SIFT et l'ORB donnent des statistiques approximativement identiques, démarrant à 80% dans le top1 et atteignant les 98% dans le top5. SFOP donne des résultats médiocres relativement aux autres. Les statistiques se situent entre 50 et 70% dans le top1. Elles n'atteignent pas les 90% même dans le top5.
Le SURF affiche une meilleure invariance à la rotation qu'à la luminosité avec des statistiques comprises entre 60 et 90%. Ceci n'empêche que ses résultats restent moins bons que ceux du SIFT et de ORB.

3.5.3 Bruit

Contrairement à la rotation et à la luminosité, l'ORB donne des résultats largement meilleurs que ceux du SIFT, du SURF et du SFOP. Il affiche des pourcentages compris entre 75 et 90% dans le top1 et le top2 et un minimum de 90% dans le top5. Ceci indique son invariance au bruit.
Les performances du SIFT et du SURF sont similaires. Ils atteignent les 80% une seule fois dans le top1 et le top2.
Malgré quelques croisements des statistiques du SFOP avec celles du SIFT et du SURF, ses résultats sont inférieurs et insuffisants par rapport à ceux des autres

3.5.4 Qualité

Globalement, les algorithmes étudiés réagissent mal à une mauvaise qualité de l'image. Le SIFT et le SFOP donne les pires résultats, qui descendent en flèche avec la diminution de la qualité de l'image. Cependant, dans le top5, les statistiques du SFOP sont légèrement meilleures que celles du SIFT.
Dans le top1 et le top2, ORB donne de meilleurs résultats que le SURF, mais n'atteint néanmoins pas les 90%.
Dans le top5, le SURF et l'ORB donnent des résultats similaires avec un minimum de 80%.

3.5.5 Echelle

Une fois de plus, le SURF affiche de moins bons résultats que le SIFT et ORB. Ses pourcentages démarrent à partir de 50% à peine, et n'atteignent la barre des 90% qu'une seule fois.
Le SIFT et l'ORB donnent de résultats satisfaisants et identiques en général. Dans le top5, les pourcentages varient entre 93 et 99%.
Le SFOP donne des résultats médiocres. Cependant, les courbes du SURF et du SFOP se confondent dans l'intervalle [0.75 ; 1.5] de l'échelle.

3.5.6 Occultations

Le SURF et le SFOP sont nettement moins invariants par rapport aux occultations, que le SIFT et l'ORB. Ils donnent des courbes pratiquement confondues et leurs pourcentages descendent en flèche en augmentant l'occultation. Ils donnent 5% de reconnaissance, pour occultation de 70% de l'image.
Le SIFT et l'ORB donnent des résultats quasiment identiques. Leurs pourcentages diminuent raisonnablement en augmentant l'occultation de l'image. Cependant, ils démarrent à 55% de reconnaissance dans le top1, pour 70% de l'occultation de l'image : Un pourcentage relativement mauvais.

3.5.7 Conclusion

Globalement, l'ORB et le SIFT donnent des résultats quasi similaires. L'ORB affiche de meilleures performances que le SIFT dans le cas de l'invariance au bruit, ce qui est en

accord avec la théorie.
En étudiant l'influence de la variation de la qualité, nous avons remarqué que le SIFT donnait de très mauvais résultats.
Le SURF donne de moins bons résultats que le SIFT et ORB en général. Ceci était prédictible, car les auteurs du SURF ont favorisé la rapidité du traitement à la qualité.
Les statistiques du SFOP étaient proches de celles du SURF à quelques reprises, comme dans le cas de la luminosité et de l'occultation. Ses résultats restent cependant médiocres relativement à ceux des autres.
Lors de l'étude de l'influence des occultations de l'image sur la reconnaissance, nous avons remarqué que les résultats obtenus étaient remarquables. Ceci car, même avec 70% d'occultation, le SIFT et ORB reconnaissent presque 60% des modèles.

Conclusion générale

La vision par ordinateur est un domaine complexe. Ses principales problématiques sont nombreuses. On peut en citer la reconnaissance de forme, la détection de mouvements, le suivi des cibles sur des vidéos etc.
A travers ce projet, nous avons découvert et étudié un ensemble d'algorithmes de traitement d'images. Ce sont des algorithmes conçus pour la reconnaissance d'objet.L'idée était de comparer pratiquement les algorithmes SIFT, SURF, SFOP et ORB.
Nous nous sommes tout d'abord intéressés à l'étude des différentes approches suivies lors de la reconnaissance d'objets. Puis, nous avons situé les algorithmes étudiés parmi ceux qui adoptent une approche locale, qui existent dans la littérature.
Nous avons ensuite étudié ces algorithmes théoriquement, de manière approfondie. Nous nous sommes ainsi familiarisés avec les différents outils mathématiques utilisés.
Dans le dernier chapitre, nous avons expliqué la procédure suivie lors de l'évaluation de ces méthodes. Nous avons ainsi énuméré les différents critères de comparaison qui sont : l'invariance aux variations de l'échelle, de la luminosité, du bruit, de la qualité de l'image. Nous avons aussi étudié les performances de ces algorithmes vis-à-vis des variations de la rotation et des occultations de l'objet sur l'image. Les résultats de ces évaluations ont été présentés sous formes de statistiques, représentant les pourcentages de reconnaissance. Nous les avons donc calculé pour le Top1, Top2 et Top5. Nous avons ainsi obtenu de réelles conclusions quant aux performances des algorithmes étudiés.
Parmi celles-ci, nous citerons les plus importantes. SIFT et ORB donnent des résultats nettement meilleurs que ceux du SURF et du SFOP. Cependant, ORB se démarque davantage.
Les perspectives de ce projet sont nombreuses. Nous proposons par exemple une implémentation de l'ORB sur un circuit programmable. Cela permettrait d'installer un système de sécurité fiable à l'entrée d'un parking ou sur une autoroute.

Bibliographie

[1] Eric NOWAK. *thèse de doctorat Reconnaissance de catégories d'objets et d'instances d'objets à l'aide de représentations locales.* PhD thesis, Institut National Polytechnique De Grenoble, 2008.

[2] Guray ERUS. *Reconnaissance d'objets cartographiques dans les images satellitaires à haute résolution.* PhD thesis, université Paris Descartes, 2008.

[3] DanielP. Huttenlocher and Shimon Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2) :195–212, 1990. ISSN 0920-5691. doi : 10.1007/BF00054921. URL http://dx.doi.org/10.1007/BF00054921.

[4] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 586–591, 1991. doi : 10.1109/CVPR.1991.139758.

[5] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(5) :530–535, 1997. ISSN 0162-8828. doi : 10.1109/34.589215.

[6] M. Delalandre et Jean-Yves Ramel T.-A. Pham, S. Barrat. Une approche robuste pour la détection de points d'intérêt dans les images de documents techniques. *Actes du Colloque International Francophone sur l'Ecrit et le Document (CIFED 2012)*, 2012.

[7] Marco Alexander Treiber. *An Introduction to Object Recognition : Selected Algorithms for a Wide Variety of Applications.* Springer Publishing Company, Incorporated, 1st edition, 2010. ISBN 1849962340, 9781849962346.

[8] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2) :151–172, 2000. ISSN 0920-5691. doi : 10.1023/A:1008199403446. URL http://dx.doi.org/10.1023/A%3A1008199403446.

[9] Karl Rohr. Recognizing corners by fitting parametric models. *International Journal of Computer Vision*, 9(3) :213–230, 1992. ISSN 0920-5691. doi : 10.1007/BF00133702. URL http://dx.doi.org/10.1007/BF00133702.

[10] Haruo Asada and Michael Brady. The curvature primal sketch. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(1) :2–14, 1986. ISSN 0162-8828. doi : 10.1109/TPAMI.1986.4767747.

[11] Karl Rohr. Recognizing corners by fitting parametric models. *International Journal of Computer Vision*, 9(3) :213–230, 1992. ISSN 0920-5691. doi : 10.1007/BF00133702. URL http://dx.doi.org/10.1007/BF00133702.

[12] L. Parida, D. Geiger, and R. Hummel. Junctions : detection, classification, and reconstruction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(7) :687–698, 1998. ISSN 0162-8828. doi : 10.1109/34.689300.

[13] Manuel Grand-brochier. *Descripteurs 2D et 2D+t de points d'intérêt pour des appariements robustes*. PhD thesis, Université Blaise Pascal - Clermont II, 2011.

[14] David.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999. doi : 10.1109/ICCV.1999.790410.

[15] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2) :91–110, November 2004. ISSN 0920-5691. doi : 10.1023/B:VISI.0000029664.99615.94. URL http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[16] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. Brief : Computing a local binary descriptor very fast. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7) :1281–1298, 2012. ISSN 0162-8828. doi : 10.1109/TPAMI.2011.222.

[17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb : An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571, 2011. doi : 10.1109/ICCV.2011.6126544.

[18] W. Förstner, T. Dickscheid, and F. Schindler. Detecting interpretable and accurate scale-invariant keypoints. In *12th IEEE International Conference on Computer Vision (ICCV'09)*, Kyoto, Japan, 2009.

[19] Andrea Vedaldi. "code - sift for matlab". http ://www.vlfeat.org/ vedaldi/code/-sift.html, consulté en Décembre 2012.

[20] Dirk-Jan Kroon. "opensurf". httphttp ://www.mathworks.com/matlabcentral/ fileexchange/28300-opensurf-including-image-warp, consulté en Décembre 2012.

[21] Kota Yamaguchi. "code - orb for matlab". http ://www.cs.stonybrook.edu/ kyamagu/mexopencv/, consulté en Avril 2013.

[22] W. Förstner, T. Dickscheid, and F. Schindler. "sfop keypoint detector". http ://www.ipb.uni-bonn.de/sfop/, consulté en Juin 2013.