

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة أبي بكر بلقايد- تلمسان
Université Aboubakr Belkaïd- Tlemcen –
Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme de MASTER**

En : Télécommunication

Spécialité : Réseaux et Système de Télécommunication

Par : MAHI Kheireddine et TRARI MEDJAOUI Nabil

Sujet

**Contribution à la détection faciale à l'aide du descripteur
AKAZE**

Soutenu publiquement, le 08 / Juin / 2017 , devant le jury composé de :

M BENDIMERAD. F.T	Professeur	Univ. Tlemcen	Président
M DERRAZ. F	Maitre de Conférences	Univ. Tlemcen	Directeur de mémoire
M BOUACHA. A	Maitre de Conférences	Univ. Tlemcen	Directeur de mémoire
M BOUABDELLAH.R	Maitre-Assistant	Univ. Tlemcen	Examineur
M ABDELLAOUI. Gh	Maitre-Assistant	Univ. Tlemcen	Examineur

Année Universitaire 2016/2017

Dédicaces

Je dédie ce modeste travail à :

*À la personne la plus cher au monde, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, pour toi **mon père**.*

*À la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur ; **maman** que j'adore.*

*Aux personnes dont j'ai bien aimé la présence dans ce jour, à Mon frère **Mohammed Réda**, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.*

À mes deux chères sœurs.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures,

*À mon binôme **Nabil** et toute la famille **Mahi**, à mes collègues d'étude **Bilal, Anes, Zoheir** et **Ismat**,
et frère de cœur, toi **Hassan**.*

Et à tous ceux qui m'aiment

KHEIREDDINE

Dédicaces

Je dédie ce modeste travail à :

*A la personne la plus chère ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi **mon père**.*

*A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur ; **maman** que j'adore.*

Aux personnes dont j'ai bien aimé la présence dans ce jour, à Mes deux frères Djallal et Abdou, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.

A mes deux chères sœurs Nabahat et Aya.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures,

*A mon binôme **Kheiro** et toute la famille **TRARI MEDJAOUI**,
à mes amis Omar, Mohammed, Wahid tks, Younese, Mnanouk,
Abdelkrim, Naïm, Mouad, Samir, Walid et Lotfi
Et mes amies Rania et les 4 Sihem.*

Et à tous ceux qui m'aiment

Nabil

Remerciement

On remercie Dieu le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

*Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de **M. F. DERRAZ**, on le remercie pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.*

*Nos remerciements s'adressent à **Mr S.M. MERIAH** professeur à l'Université Abou Bekr Belkaid de Tlemcen pour son aide pratique et son soutien moral et ses encouragements.*

Nos remerciements s'adressent également à tous nos professeurs de la spécialité RST pour leurs générosités et la grande patience dont ils ont su faire preuve malgré leurs charges académiques et professionnelles.

Enfin, il nous serait impossible de terminer ces remerciements sans adresser une pensée chaleureuse à tous nos amis et à toutes les personnes qui nous ont aidées de près ou de loin.

***TABLE DES
MATIERES***

TABLE DES MATIÈRES

Dédicaces.....	i
Remerciement.....	iii
Introduction générale.....	1

Chapitre I : Généralités sur les méthodes de détection de primitive

I.1	Introduction.....	2
I.2	Apprentissage Automatique.....	2
I.2.1	Introduction.....	2
I.2.2	Principes.....	3
I.2.3	Domaines de l'Apprentissage Automatique.....	3
I.2.4	Types d'apprentissage automatique.....	4
I.3	Machine Learning et la détection de visage.....	6
I.3.1	Détection de visage.....	6
I.3.1.1	Machine à Vecteurs de Support	6
I.3.1.2	Scale-Invariant Features Transform	8
I.3.1.3	Kaze	8
I.3.1.4	Speeded Up Robust Features	10
I.4	Conclusion.....	10

Chapitre II : Détection de visage par SIFT et AKAZE

II.1	Introduction.....	11
II.1.1	Détection par SURF.....	24
II.1.1.1	Détecteur Fast-Hessian	25
II.1.1.2	Le descripteur SURF.....	26

II.1.1.3	Les limitations du SIFT et SURF.....	27
II.1.2	Détection par algorithme AKAZE.....	28
II.1.3	Descripteur KAZE	28
II.1.3.1	Introduction	28
II.1.3.2	La description par AKAZE.....	29
II.1.3.3	Caractéristiques KAZE	29
II.1.4	Détection des features par AKAZE	30
II.1.5	Détection de visage par descripteur AKAZE	30
II.2	Algorithme de détection.....	31
II.2.1	Algorithme AKAZE.....	31
II.2.2	Présentation d'AKAZE.....	31
II.3	Conclusion.....	32

Chapitre III : Application de descripteur AKAZE

III.1	Introduction à la bibliothèque OpenCV	33
III.2	La bibliothèque QT	33
III.3	Les paramètres du descripteur AKAZE	33
III.3.1	Traitement d'images.....	34
III.3.2	Traitement des vidéos.....	34
III.3.3	Le moteur de production « CMake ».....	34
III.4	Application du descripteur AKAZE.....	35
III.5	La construction du descripteur AKAZE.....	38
III.6	Discussion des résultats	39
III.7	Conclusion	39
	Conclusion générale.....	II

LISTE DES FIGURES

LISTE DES FIGURES

Figure I.1. Schéma d'un modèle supervisé.	4
Figure I.2 Schéma d'un modèle non supervisé.	5
Figure I.3. Schéma d'un modèle semi-supervisé ou incrémental.....	5
Figure I.4 Principe du Séparateur à Vaste Marge (SVM).	7
Figure I.5 Hyperplan Optimal et Marge	7
Figure I.6 Exemple de résultat de la comparaison de deux images par la méthode SIFT	8
Figure I.7 Flou avec la Différence de Gaussien (à gauche). Flou avec filtrage de diffusion non linéaire (à droite).....	9
Figure II.1 Pyramide de gradients : 3 octaves de 5 gradients.....	12
Figure II.2 Construction de la pyramide de différences de gaussiens (DoG) à partir de la pyramide de gradients.	13
Figure II.3 Détection des caractéristiques dans l'espace d'échelle.....	14
Figure II.4 Construction de l'histogramme des orientations.....	20
Figure II.5 Construction d'un descripteur SIFT.	21
Figure II.6 De gauche à droite: les dérivées partielles gaussiennes (discrétisées et recadrées) en direction de y et en direction de xy , et nos approximations en utilisant des filtres à caisse. Les régions grises sont égales à zéro.....	26
Figure II.7 Points d'intérêt détectés pour un champ de tournesol.....	27
Figure II.8 LDB et les tests binaires M-LDB proposés entre les divisions de grille autour d'un point clé, montrés pour l'intensité et les gradients dans x . M-LDB comprend la rotation et le sous-échantillonnage qui dépend de l'échelle.....	30
Figure III.1 Exemple d'une séquence vidéo dans un studio.....	35
Figure III.2 Exemple d'une séquence vidéo sur une scène	36
Figure III.3 Exemple d'une séquence vidéo d'un film.....	36
Figure III.4 Exemple d'une séquence vidéo en plein aire.	37
Figure III.5 Exemple d'une séquence vidéo dans un lieu fermé.....	37
Figure III.6 Exemple d'observation en full.	37

LISTE DES FIGURES

Figure I.1. Schéma d'un modèle supervisé.....	4
Figure I.2. Schéma d'un modèle non supervisé.....	5
Figure I.3. Schéma d'un modèle semi-supervisé ou incrémental.....	5
Figure I.4. Principe du Séparateur à Vaste Marge (SVM).....	7
Figure I.5. Hyperplan Optimal et Marge.....	7
Figure I.6. Exemple de résultat de la comparaison de deux images par la méthode SIFT.....	8
Figure I.7. Flou avec la Différence de Gaussien (à gauche). Flou avec filtrage de diffusion non linéaire (à droite).....	9
Figure II.1. Pyramide de gradients : 3 octaves de 5 gradients.....	12
Figure II.2. Construction de la pyramide de différences de gaussiens (DoG) à partir de la pyramide de gradients.....	13
Figure II.3. Détection des caractéristiques dans l'espace d'échelle en utilisant une pyramide de gaussienne (à gauche) les niveaux voisins d'une pyramide gaussienne sub-octave sont soustraits pour produire des images de différences de gaussienne.....	14
Figure II.4. Construction de l'histogramme des orientations.....	17
Figure II.5. Construction d'un descripteur SIFT.....	18
Figure II.6. De gauche à droite: les dérivées partielles gaussiennes (discrétisées et recadrées) en direction de x et en direction de y , et nos approximations en utilisant des filtres à caisse. Les régions grises sont égales à zéro.....	21
Figure II.7. Points d'intérêt détectés pour un champ de tournesol.....	22
Figure II.8. LDB et les tests binaires M-LDB proposés entre les divisions de grille autour d'un point clé, montrés pour l'intensité et les gradients dans x . M-LDB comprend la rotation et le sous-échantillonnage qui dépend de l'échelle.....	25
Figure III.1. Exemple d'une séquence vidéo dans un studio.....	30
Figure III.2. Exemple d'une séquence vidéo sur une scène.....	31
Figure III.3. Exemple d'une séquence vidéo d'un film.....	31

Figure III.4. Exemple d'une séquence vidéo en plein aire.....	32
Figure III.5. Exemple d'une séquence vidéo dans un lieu fermé.....	32
Figure III.6. Exemple d'observation en full.....	32

INTRODUCTION
GENERALE

INTRODUCTION GÉNÉRALE

Les descripteurs locaux sont le résultat de deux étapes. Dans un premier temps, des points d'intérêt dans l'image sont détectés. Ensuite, pour chaque point d'intérêt, un descripteur qui décrit localement l'image autour du point est construit. Par la suite, une description de l'image est donnée à partir d'une description de certaines zones de cette même image.

L'objectif est de retrouver les mêmes zones décrites de façon similaire même si l'image subit des déformations. La qualité d'un détecteur de points d'intérêt se mesure par sa répétabilité.

C'est-à-dire que l'on doit idéalement retrouver les mêmes points d'intérêt après que l'image a subi une modification. Un descripteur sera d'autant meilleur qu'il différencie les zones locales décrites. Il doit aussi être robuste par rapport aux changements possibles (illumination, angle de vue, échelle, rotation...) et aussi à de petites variations de la position du point d'intérêt. Si l'image est déformée ou si la photo est prise d'un autre point de vue, on retrouverait certainement quasiment les mêmes points. Dans les paragraphes suivants, nous présentons rapidement les différents types de détecteurs de points d'intérêt et de descripteurs. Ensuite, l'algorithme SIFT (pour Scale Invariant Features Transform) ainsi que son extension AKAZE que nous avons choisi est présenté en détail (il regroupe un détecteur de régions d'intérêt ainsi qu'un descripteur).

Notre mémoire est organisé comme suit :

Dans le prochain chapitre nous allons présenter sommairement les méthodes d'apprentissage et nous finirons par une présentation des descripteurs discrets notamment le SIFT, SURF et le AKAZE.

Dans le deuxième chapitre, nous avons présenté en détails le descripteur SIFT et son dérivée, à savoir le descripteur AKAZE dans le cadre de détection de visage.

Dans le chapitre 3 nous avons présenté quelques résultats essentiels par rapport à l'utilisation du descripteur AKAZE.

C

H

A

P

I

T

R

E

I

**GENERALITES SUR LES
METHODES DE DETECTION
DE PRIMITIVE**

Chapitre I : Généralités sur les méthodes de détection de primitive

I.1 Introduction

La vision par ordinateur est une branche de l'intelligence artificielle dont le principal but est de permettre à une machine d'analyser, traiter et comprendre une image ou flux vidéo pris par un système d'acquisition. Cette approche a connu un intérêt grandissant grâce au développement de la vidéosurveillance et à la démocratisation des appareils photos numériques, des téléphones mobiles et des réseaux permettant l'échange des images entre différents supports [1].

Dans ce vaste domaine, nous sommes intéressés à la reconnaissance d'objets, et plus particulièrement à la détection des visages. Cette dernière combine des approches de traitement d'images et d'apprentissage automatique dans le but de mener à bien cette tâche de détection des visages dans une scène vidéo.

Toutefois, dans ce qui suit, nous allons commencer à présenter sommairement quelques algorithmes d'apprentissage automatique qui ont été utilisés dans le domaine de la détection de visage.

I.2 Apprentissage Automatique

I.2.1 Introduction

L'apprentissage automatique ou apprentissage statistique (*machine learning*), champ d'étude de l'intelligence artificielle, concerne la conception, l'analyse, le développement et l'implémentation de méthodes permettant à une machine (au sens large) d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou problématiques à résoudre par des algorithmiques classiques (classification homme/femme) [1].

L'analyse peut concerner des graphes, arbres, ou courbes (par exemple, la courbe d'évolution temporelle d'une mesure ; on parle alors de données continues, par opposition aux données discrètes associées à des attributs-valeurs classiques) au même titre que de simples nombres.

L'apprentissage automatique consiste à utiliser des calculateurs/ordinateurs pour optimiser un modèle de traitement de l'information selon certains critères de performance à partir d'observations, que ce soit des données-exemples ou des expériences passées [2].

I.2.2 Principes

Par l'entraînement sur un grand nombre de mesures, il devient facile à un programme d'apprentissage automatique de reconnaître des formes, même complexes, et d'y classer ensuite de nouveaux points.

La majorité des algorithmes d'apprentissage automatique permettent à un système piloté par ordinateur (un robot éventuellement), ou assisté par ordinateur, d'adapter ses analyses et ses comportements en réponse, en se fondant sur l'analyse de données empiriques provenant d'une base de données ou de capteurs.

La difficulté réside dans le fait que l'ensemble de tous les comportements possibles compte tenu de toutes les entrées possibles devient rapidement trop complexe à décrire (on parle d'explosion combinatoire). On confie donc à des programmes le soin d'ajuster un modèle pour simplifier cette complexité et de l'utiliser de manière opérationnelle. Idéalement, l'apprentissage visera à être non supervisé ou la nature des données d'entraînement n'est pas connue [1].

Ces programmes, selon leur degré de perfectionnement, intègrent éventuellement des capacités de traitement probabiliste des données, d'analyse de données issues de capteurs, de reconnaissance (reconnaissance vocale, reconnaissance de forme, d'écriture...) [3], de data-mining [4], d'informatique théorique [5], etc.... .

I.2.3 Domaines de l'Apprentissage Automatique [1]

Les principaux domaines d'applications de l'apprentissage automatique (AA) sont les fouilles de données et l'intelligence artificielle.

- La fouille de données (« Data Mining ») est le processus d'extraction de la connaissance : il consiste à sélectionner les données à étudier à partir de Bases de Données (BDs) (hétérogènes ou homogènes), à épurer ces données et enfin à les utiliser en apprentissage pour construire un modèle.

Exemples :

- Trouver une prescription pour un malade (patient) à travers des fichiers médicaux antérieurs.
- Apprentissage de la reconnaissance de transactions frauduleuses par carte de crédit, par examen des transactions passées avérées frauduleuses.

- L'intelligence artificielle, la vision par ordinateur, la robotique, l'analyse et la compréhension des images, la reconnaissance de formes, reconnaître des objets dans les vidéo et extraire des contenus sémantiques des images sont autant d'applications qui requièrent la construction de modèles par apprentissage automatique [2].

Exemples :

- Systèmes de vidéo surveillance pour la détection des intrus.
- Logiciel biométrique de reconnaissance de visages et d'empreintes digitales.

I.2.4 Types d'apprentissage automatique

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage utilisé

❖ Apprentissage Supervisé :

A pour objectif de concevoir un modèle reliant des données d'apprentissage à un ensemble de valeurs de sortie (Fig I.1). Si les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classement ; on parle alors d'apprentissage supervisé (ou d'analyse discriminante) [1]. Dans ce cas, un expert doit préalablement étiqueter des exemples. Par la suite, le processus se passe en deux phases. Lors de la première phase (hors ligne, dite d'apprentissage), il s'agit de déterminer un modèle des données étiquetées. La seconde phase (en ligne, dite de test) consiste à prédire l'étiquette d'une nouvelle donnée, connaissant le modèle préalablement appris. Parfois il est préférable d'associer une donnée non pas à une classe unique, mais une probabilité d'appartenance à chacune des classes prédéterminées (on parle alors d'apprentissage supervisé probabiliste) [1,2].

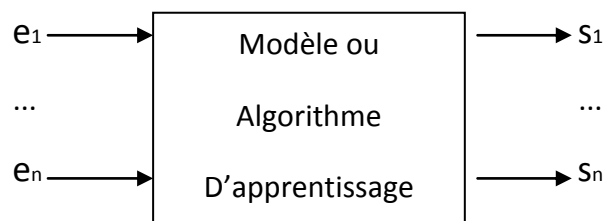


Figure I.1 Schéma d'un modèle supervisé.

❖ Apprentissage Non-Supervisée :

Vise à concevoir un modèle structurant l'information. La différence ici est que les comportements(ou catégories ou encore les classes) des données d'apprentissage ne sont pas connus, c'est ce que l'on cherche à trouver [3] (Figure I.2).

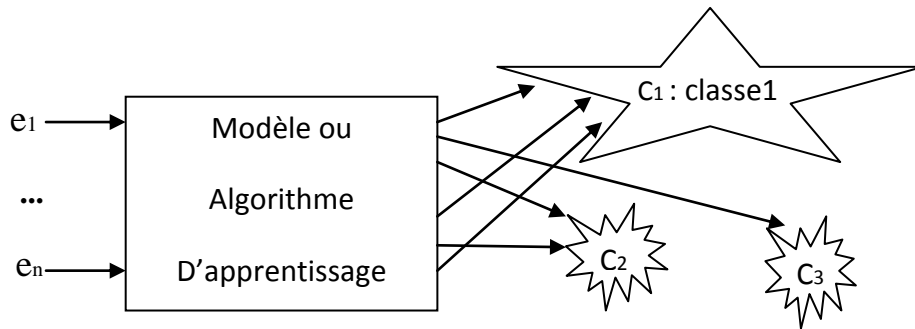


Figure I.2 Schéma d'un modèle non supervisé.

Quand le système ou l'opérateur ne disposent que d'exemples non étiquetés, et que le nombre de classes et leur nature n'ont pas été prédéterminés, on parle d'apprentissage non supervisé ou clustering [6]. Aucun expert n'est requis. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données. Le partitionnement de données (data clustering), est un algorithme d'apprentissage non supervisé.

❖ Apprentissage par Renforcement :

Les données en entrée sont les mêmes que pour l'apprentissage supervisé, cependant l'apprentissage est guidé par l'environnement sous la forme de récompenses ou de pénalités données en fonction de l'erreur commise lors de l'apprentissage [4].

❖ Apprentissage Semi-Supervisé

Les données d'entrée sont constituées d'exemples étiquetés et non étiquetés (Fig I.3). Ce qui peut être très utile quand on a deux types de données, car cela permet de ne pas en laisser de côté et d'utiliser toute l'information [5].

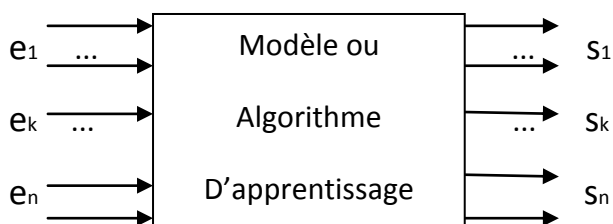


Figure I.3 Schéma d'un modèle semi-supervisé ou incrémental.

I.3 Machine Learning et la détection de visage

Dans les deux dernières décennies, le problème de détection et de reconnaissance de visage est devenu un problème très attrayant par le nombre important des défis rencontrés. Plusieurs solutions sont proposées dans la littérature en particulier les solutions basées sur des techniques d'apprentissage automatique [6], que nous allons les présenter par la suite.

I.3.1 Détection de visage

Les visages constituent une catégorie de stimulus importante par la richesse des informations qu'ils véhiculent. Ils sont à la fois des vecteurs visuels principaux à l'identité des individus et des vecteurs essentiels de communication (verbale et non verbale) [6]. La détection automatique de visages est un problème très important. Du point de vue application, elle est à la base de tous systèmes de reconnaissance de visages, car avant de reconnaître n'importe quelle personne il est indispensable de localiser son visage. La détection de visages dans une image sans hypothèse préalable est un problème très difficile à résoudre en raison de la très grande variabilité de la forme et la couleur à détecter (Orientation et tailles quelconques, éclairage variable, couleur de peau différente) [7].

Il existe plusieurs techniques (ou algorithmes) pour la détection de visage ; basées sur des techniques d'apprentissage automatique, qui créent des modèles après la phase d'apprentissage et puis les utilisent pour trouver des visages [7] ; parmi ces algorithmes on décrit notamment : SVM [8], SIFT [7], SURF [7], AKAZE [8].

I.3.1.1 Machine à Vecteurs de Support

Une machine à vecteurs de support ou séparateurs à vastes marges (notées SVM pour *Support Vector Machines*), est une méthode d'apprentissage supervisé utilisé pour la classification et la régression. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle [9].

➤ Le principe de SVM

Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan. Dans le schéma qui suit (Fig I.4), on détermine un hyperplan qui sépare les deux ensembles de points [8].

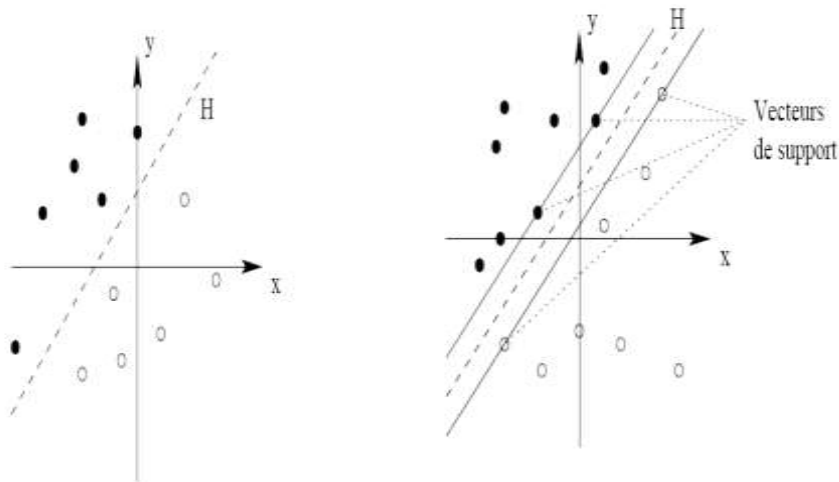


Figure I.4 Principe du Séparateur à Vaste Marge (SVM).

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support. Nous allons en plus chercher parmi les hyperplans valides, celui qui passe «au milieu» des points des deux classes d'exemples, qu'on va l'appeler «Hyperplan Optimal» (Fig I.5), dont la distance minimale aux exemples d'apprentissage est maximale. On appelle cette distance « marge » entre l'hyperplan et les exemples [8]. Comme on cherche à maximiser cette marge, on parlera de *séparateurs à vaste marge*.

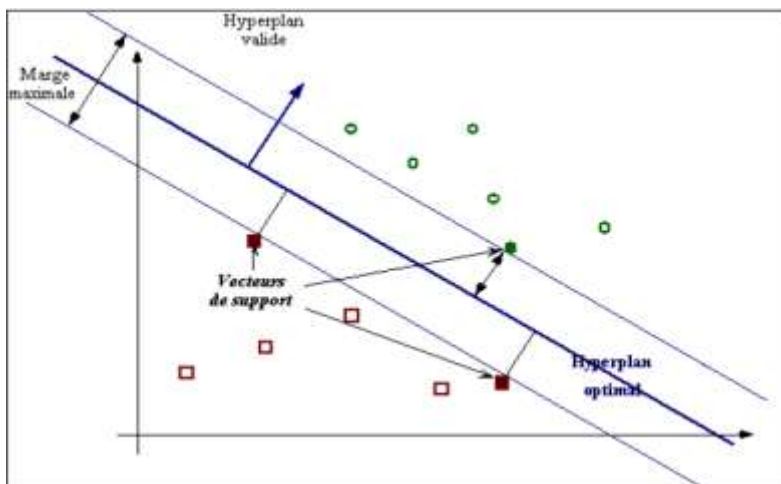


Figure I.5 Hyperplan Optimal et Marge.

Les entrées x sont transformées en un vecteur dans un espace de Hilbert F . Soit x le vecteur associé. On définit $f(x) = \omega x + \beta$ et l'hyperplan a pour équation $\omega x + \beta = 0$. La distance d'un point au plan est donnée par :

$$d(x) = |\omega x + \beta| \div \|\omega\| \quad (1.1)$$

Le classement est correct si $yf(x) > 0$ ou à un coefficient près $yf(x) \geq 1$.

Maximiser la marge revient à minimiser $\|\omega\|$ ou $\|\omega\|^2 / 2$ sous les contraintes $y_i f(x_i) \geq 1$.

On utilise la méthode des multiplicateurs de Lagrange en ne conservant que les vecteurs x_i les plus proches de l'hyperplan (vecteurs supports). Lorsque tous les cas ne sont pas séparables, on introduit un terme d'erreur :

$$y_i f(x_i) \geq 1 - \varepsilon_i \quad (1.2)$$

La transformation en vecteur ne fait intervenir que l'expression du produit scalaire dans F . On recherche en fait directement l'expression du produit scalaire à partir des coordonnées initiales à l'aide d'une fonction k appelée noyau. On distingue les noyaux linéaire, polynomiaux, gaussien... s'adaptant aux différentes problématiques rencontrées [1].

I.3.1.2 Scale-invariant feature transform (SIFT)

Le descripteur SIFT est défini comme « transformation de caractéristiques visuelles invariante à l'échelle », est un algorithme utilisé dans le domaine de la vision par ordinateur pour détecter et identifier les éléments similaires entre différentes images numériques (éléments de paysages, objets, personnes, etc.). Il a été développé en 1999 par le chercheur David Lowe [6].

L'étape fondamentale de la méthode proposée par Lowe consiste à calculer ce que l'on appelle les « descripteurs SIFT » des images à étudier (Fig I.6). Il s'agit d'informations numériques dérivées de l'analyse locale d'une image et qui caractérisent le contenu visuel de cette image de la façon la plus indépendante possible de l'échelle (« zoom » et résolution du capteur), du cadrage, de l'angle d'observation et de l'exposition (luminosité).



Figure I.6 Exemple de résultat de la comparaison de deux images par la méthode SIFT.

I.3.1.3 KAZE

L'algorithme KAZE a été développé en 2012 et il est dans le domaine public. Le nom fait référence au flux d'air régi par des processus non linéaires à grande échelle. L'idée derrière

la création de cet algorithme était de détecter et de décrire les fonctionnalités 2D dans une échelle non linéaire extrême espace afin d'obtenir une meilleure précision de localisation et le caractère distinctif [9]. Le flou gaussien utilisé dans d'autres algorithmes de reconnaissance d'objets, tels que SIFT, ne respecte pas les limites naturelles des objets puisque les détails de l'image et le bruit sont lissés au même degré à tous les niveaux d'échelle. Pour rendre le flou adapté aux fonctionnalités d'image, KAZE utilise le filtrage de diffusion non linéaire parallèlement à la méthode AOS (Additive Operator Splitting). Avec ce filtrage, le bruit de l'image est réduit, mais les limites de l'objet sont conservées, comme le montre la figure 6. Pour la reconnaissance d'objet, KAZE suit principalement les mêmes étapes que le SIFT mais avec des différences dans chaque étape.



Figure I.7 Flou avec la Différence de Gaussien (à gauche), Flou avec filtrage de diffusion non linéaire (à droite).

Cet algorithme utilise un filtrage de diffusion non linéaire combiné à une fonction de conductivité au lieu du noyau gaussien d'échelle utilisé dans SIFT. Cette fonction de conductivité était proposée par Perona et Malik et prend le gradient d'une version gaussienne lissée de l'image originale (∇L_σ) en fonction du temps. Le but est d'obtenir des caractéristiques qui présentent une plus grande répétabilité et un caractère distinctif que les EIPD.

$$c(x, y, t) = g(|\nabla L_\sigma(x, y, t)|) \quad (1.3)$$

Contrairement à SIFT, qui utilise la Différence de Gaussiens (DoG) pour traiter les images floues, KAZE utilise le schéma numérique AOS. KAZE utilise également l'image originale pour créer les images floues et n'effectue pas l'échantillonnage en panne dans EIPD.

➤ AKAZE

La méthode de KAZE se base essentiellement sur la résolution d'une équation aux dérivées partielles (EDP) du filtre à diffusion anisotrope. Cela ne peut pas être fait forçant analytiquement KAZE à utiliser le schéma AOS pour résoudre les EDP. Cependant, ce processus est coûteux en termes de calcul et, par conséquent, une version accélérée de KAZE a été créée. Cette version s'appelle Accelerated KAZE ou AKAZE.

L'algorithme KAZE accéléré fonctionne de la même manière que KAZE, mais contrairement à KAZE, il utilise une méthode plus rapide pour créer l'espace d'échelle non-linéaire pour la diffusion rapide et explicite (FED). Le FED est une décomposition des filtres à caisse et utilise l'image originale pour le premier cycle, l'image résultante est utilisée pour le cycle suivant.

En plus de l'utilisation des FED, AKAZE utilise un descripteur binaire pour augmenter encore la vitesse. Pour créer la description du point clé, AKAZE utilise une version modifiée du descripteur Binary Local Difference Binary (LDB) [34].

I.3.1.4 Speeded Up Robust Features (SURF)

Speeded Up Robust Features que l'on peut traduire par caractéristiques robustes accélérées, est un algorithme de détection de caractéristique et un descripteur, présenté par des chercheurs de l'ETH Zurich et de la Katholieke Universiteit Leuven pour la première fois en 2006 puis dans une version révisée en 2008. Il est utilisé dans le domaine de vision par ordinateur, pour des tâches de détection d'objet ou de reconstruction 3D.

SURF est partiellement inspiré par le descripteur SIFT, qu'il surpasse en rapidité et, selon ses auteurs, plus robuste pour différentes transformations d'images. SURF est fondé sur des sommes de réponses d'ondelettes de Haar 2D et utilise efficacement les images intégrales. En tant que caractéristique de base, SURF utilise une approximation d'ondelettes de Haar du détecteur de blob à base de déterminant Hessien [10].

Il est utilisée car les descripteurs doivent fonctionner dans un modèle d'espace échelle non linéaire. Les réponses dérivées sont calculées et résumés dans un vecteur descripteur de points clés, puis le vecteur est centré au point clé. Enfin, le descripteur est normalisé en un vecteur unitaire.

Étant donné que KAZE calcule des dérivées multi-échelles (gradients) pour chaque pixel, il est plus coûteux de calculer que SURF, bien que toujours comparable à SIFT, mais il enregistre les efforts de calcul de la description de point de référence car le même ensemble de dérivations est utilisé pour la description d'un point clé.

I.4 Conclusion

Nous avons vu dans ce premier chapitre, brièvement, un état de l'art sur le développement des techniques de l'apprentissage automatique, et les méthodes adoptées pour la détection de visage, ainsi que leurs principes pour la classification et la reconnaissance d'objets.

Le chapitre suivant, nous allons reprendre en détail les descripteurs locaux utilisés pour la localisation du visage. En particulier, nous passons en revue les descripteurs de la famille SIFT et d'autres descripteurs dérivés tel que le SURF et l'AKAZE.

C

H

A

P

I

T

R

E

II

**DETECTION DE VISAGE PAR
DESCRIPTEUR SIFT ET AKAZE**

Chapitre II : Détection de visage par SIFT et AKAZE

II.1 Introduction

Vers la fin des années quatre vingt dix, des chercheurs Canadiens ont présenté une nouvelle méthode pour l'extraction de caractéristique et la détection d'objets dans des images en niveaux de gris [11]. Ces derniers, ont proposé de transformer les données d'une image en coordonnées et à échelle rappelées des caractéristiques locales. Plusieurs améliorations sont énumérées par la suite [11]. Inspiré par les travaux de Hans Moravec sur l'utilisation des points d'intérêt et les travaux de Harris et Stephens sur la correspondance entre images stéréoscopiques. L'intérêt de ce type de détecteur est confirmé par les travaux de Harris [12] et le descripteur de coins qu'il propose, améliorant la plupart des défauts du détecteur de Moravec, va connaître un important succès. D'autres travaux de Cordolite et Molw ont établi l'importance dans le domaine de vision par ordinateur par la recherche des caractéristiques locales invariantes [41]. Cependant le descripteur développe s'appuie toujours sur le détecteur de Harris reste toujours sensible aux changements d'échelle, d'angle, d'observation et l'exposition. Lowe et ses collaborateurs proposent de nouveau descripteur SIFT pour y remédier à ces plusieurs. Lowe comblera grandement ces défauts avec son descripteur SIFT [11].

La méthode proposée par Lowe comprend deux parties (chacune ayant fait l'objet de recherches plus ou moins indépendantes par la suite) : un algorithme de détection de caractéristiques et de calcul de descripteurs ; un algorithme de mise en correspondance proprement dit.

De ces deux aspects, le premier est sans doute celui qui a le plus assuré la popularité de la méthode, à tel point que le sigle SIFT fait plus souvent référence aux « descripteurs SIFT » qu'à la méthodologie globale. Il s'agit tout d'abord de détecter sur l'image des zones circulaires « intéressantes », centrées autour d'un point clé et de rayon déterminé appelé facteur d'échelle. Celles-ci sont caractérisées par leur unité visuelle et correspondent en général à des éléments distincts sur l'image.

Sur chacune des échelles, on détermine une orientation intrinsèque qui va aider à construire un histogramme des orientations locales des zones (pondéré, seuillé et normalisé). Cet histogramme mis sous forme d'un vecteur à 128 dimensions (ou valeurs) constitue le descripteur SIFT du point-clé, et l'ensemble des descripteurs d'une image établissent ainsi une véritable signature numérique du contenu de celle-ci.

Ces descripteurs présentent l'avantage d'être invariants à l'orientation et à la résolution de l'image, et peu sensibles à son exposition et à sa netteté [9].

Une fois le calcul des descripteurs effectué, l'algorithme de mise en correspondance intervient dans le but de rechercher les zones d'une image dite suspecte qui contiennent des éléments visuellement similaires à ceux d'une bibliothèque d'images de référence, en d'autres termes les descripteurs numériquement proches. Ceci est dû à la fois à la robuste des descripteurs et le caractère discriminant du SIFT.

- **Détection des points-clés du SIFT** : Chaque point-clé est calculé sur des zones d'intérêt circulaire de rayon proportionnel au facteur d'échelle σ . Ce point-clé est ensuite associé à une orientation intrinsèque ne dépendant que du contenu local de l'image autour de ce même point-clé et pour le même facteur d'échelle. Ceci assure une invariance à la rotation et est utilisée comme référence dans le calcul du descripteur SIFT.
- **Détection d'extremums dans l'espace des échelles** : La détection s'effectue dans un espace discret que l'on appelle espace des échelles (Scale Space) qui comporte trois dimensions : les coordonnées cartésiennes x et y et le facteur d'échelle σ . On appelle gradient de facteur d'échelle σ (noté L) le résultat de la convolution d'une image I par un filtre gaussien G de paramètre σ [24], soit :

$$\begin{cases} L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \\ G(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \end{cases} \quad (\text{II.1})$$

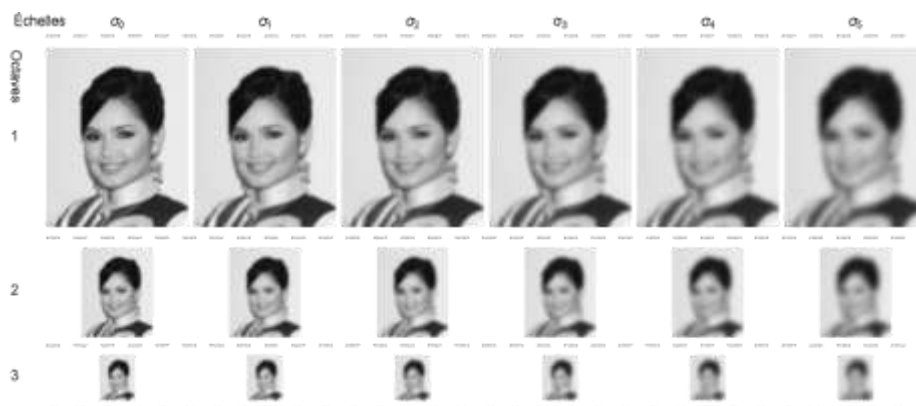


Figure II.1 Pyramide de gradients : 3 octaves de 5 gradients.

Cette convolution a pour effet de lisser l'image originale I ou les détails trop petits sont estompés. Par conséquent, la détection des objets de dimension approximativement égale à σ se fait en étudiant l'image appelée différences de gaussiennes (Difference Of Gaussians : DoG) définie par :

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (\text{II.2})$$

avec k est un paramètre constant pour fixer la finesse de la discrétisation de l'espace des échelles.

- Algorithme de DoG

Input: (v^o_s) , digital Gaussian scale-space, $o=1, \dots, n_{\text{oct}}$ and $s=0, \dots, n_{\text{spo}} + 2$.

Output: (w^o_s) , digital DoG, $o=1, \dots, n_{\text{oct}}$ and $s=0, \dots, n_{\text{spo}} + 1$.

for $o=1, \dots, n_{\text{oct}}$ and $s=0, \dots, n_{\text{spo}} + 1$ **do**

for $m=0, \dots, M_0 - 1$ and $n=0, \dots, N_0 - 1$ **do**

$w^o_s(m, n) = v^o_{s+1}(m, n) - v^o_s(m, n)$

Dans cette image ne persistent plus que les objets observables dans les octaves qui varient entre σ et $k\sigma$. De ce fait, un point-clé candidat (x, y, σ) est défini comme un point où un extremum du DoG est atteint par rapport à ses voisins immédiats (les 26 points voisins) :

$$\left\{ D(x + \delta_x, y + \delta_y, s\sigma), \delta_x \in \{-1, 0, 1\}, \delta_y \in \{-1, 0, 1\}, s \in \{k^{-1}, 1, k\} \right\} \quad (\text{II.3})$$

L'utilisation d'une pyramide est préconisée pour optimiser le temps de calcul des images floutées à des octaves différentes. Au sein d'une même octave, le nombre de convolutions à calculer est constant (voir Figure II.2).

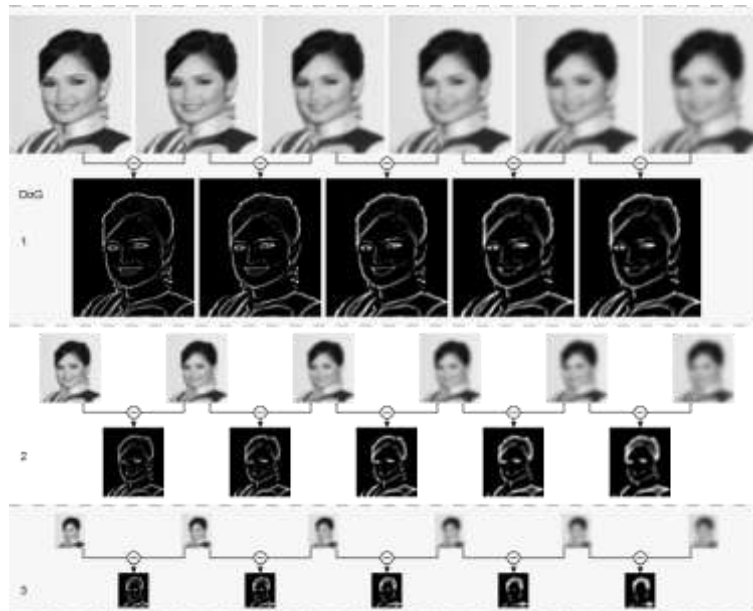


Figure II.2 Construction de la pyramide de différences de gaussiens (DoG) à partir de la pyramide de gradients.

Le facteur fixe k dans les formules ci-dessus est calculé pour qu'au final, l'espace discrétisé des facteurs d'échelles considérés corresponde à une progression géométrique $\{\sigma, k\sigma, k^2\sigma, \dots\}$, avec à chaque changement d'octave une valeur $k\sigma$ qui devient égale à une quantité de la forme $2^{2\sigma}$ (Voir Figure II.3). La progression géométrique des facteurs d'échelle – est important pour que les valeurs des DoG à différentes échelles soient comparables entre elles et à faire éviter le recours un facteur de normalisation dans calcul [24].

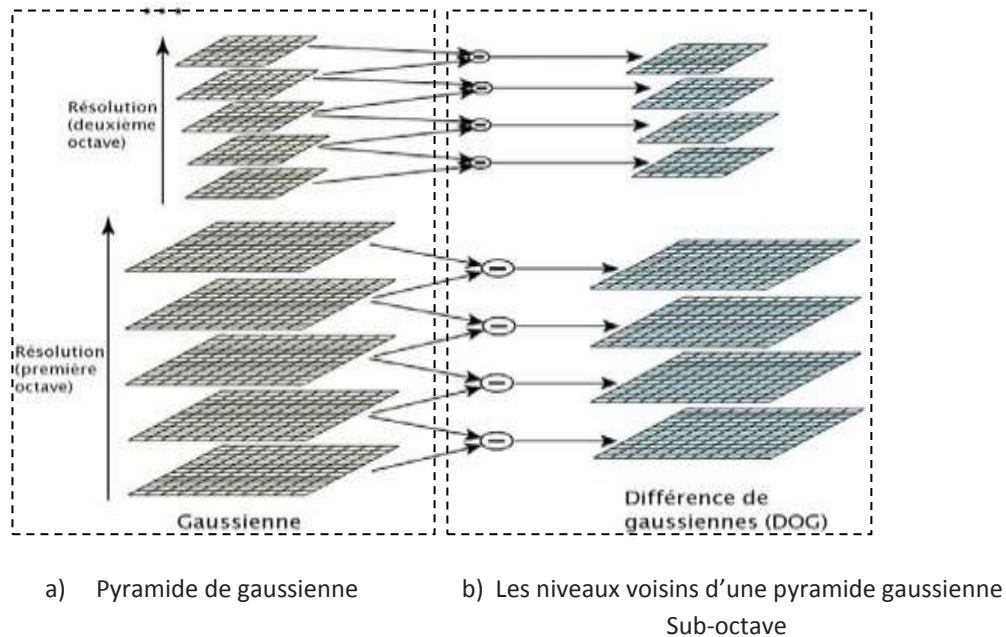


Figure II.3 Détection des caractéristiques dans l'espace d'échelle

L'étape de détection des points-clés candidats décrite ci-dessus est une variante de l'une des méthodes de blob détection [18] (détection de zones) développée par Lindeberg, qui utilise le Laplacien normalisé par le facteur d'échelle au lieu des DoG.

Ces derniers peuvent être considérés comme une approximation des laplaciens et présentent l'avantage d'autoriser l'utilisation d'une technique pyramidale.

➤ Localisation précise de points clés

L'étape de détection d'extremums produit en général un grand nombre de points-clés candidats, dont certains sont instables ; de plus, leur localisation les octaves supérieures de la pyramide. D'autres traitements sont appliqués, pour : 1) correction des positions des points clés, 2) éliminer les points clé de faible contraste ou situés sur des arêtes de contour à faible courbure et donc susceptibles de glisser facilement.

- Algorithme de points clés

Input : $l_c = \left\{ \left(o_{key}, s_{key}, x_{key}, y_{key}, \sigma_{key}, \omega \right) \right\}$, list of keypoints.

$(\theta_m V_{s,m,n}^o)$, scale space gradient along $x, o = 1, \dots, n_{oct}$ and $s = 1, \dots, n_{spo}$

$(\theta_n V_{s,m,n}^o)$, scale space gradient along $y, o = 1, \dots, n_{oct}$ and $s = 1, \dots, n_{spo}$

Parameters : λ_{ori} . The patch P^{ori} is $6\lambda_{ori}\sigma$ wide.

The Gaussian window has a standar deviation of $\lambda_{ori}\sigma$

n_{bins} , number of bins in the orientation histogram h.

T, threshold for secondary reference orientations.

Output : $l_D = \{(o, s', m', n', x, y, \sigma, \omega, \theta)\}$ list of oriented keypoints.

Temporary : h_k , orientation histogram, $k = 1, \dots, n_{bins}$ and with h_k covering

$$\left[\frac{2\pi(k-3/2)}{n_{bins}}, \frac{2\pi(k-1/2)}{n_{bins}} \right]$$

for each keypoint $(o_{key}, s_{key}, x_{key}, y_{key}, \sigma_{key}, \omega)$ in l_c **do**

//Check if the keypoint is distant enough from the image borders

If $3\lambda_{ori}\sigma \leq x_{key} \leq h - 3\lambda_{ori}\sigma$ and $3\lambda_{ori}\sigma \leq y_{key} \leq \omega - 3\lambda_{ori}\sigma$ then

//Initialize the orientation histogram h

for $1 \leq k \leq n_{bins}$ **do** $h_k \leftarrow 0$

//Accumulate samples from the normalized patch P^{ori}

for $m = \left[\left(x_{key} - 3\lambda_{ori}\sigma_{key} \right) / \delta_{o_{key}}, \dots, \left(x_{key} + 3\lambda_{ori}\sigma_{key} \right) / \delta_{o_{key}} \right]$ **do**

for $n = \left[\left(y_{key} - 3\lambda_{ori}\sigma_{key} \right) / \delta_{o_{key}}, \dots, \left(y_{key} + 3\lambda_{ori}\sigma_{key} \right) / \delta_{o_{key}} \right]$ **do**

//compute the sample contribution

$$c_{m,n}^{ori} = e^{-\frac{\left\| (m\delta_{o_{key}}, n\delta_{o_{key}}) - (x_{key}, y_{key}) \right\|^2}{2(\lambda_{ori}\sigma_{key})^2}} \left\| \left(\theta_m v_{s_{key}}^{o_{key}}, m, n, \theta_n v_{s_{key}}^{o_{key}}, m, n \right) \right\|$$

//compute the corresponding bin index

$$b_{m,n}^{ori} = \left[\frac{n_{bins}}{2\pi} \left(\arctan 2 \left(\theta_m v_{s_{key}}^{o_{key}}, m, n, \theta_n v_{s_{key}}^{o_{key}}, m, n \right) \bmod 2\pi \right) \right]$$

//Update the histogram

$$h_{b_{m,n}^{ori}} \leftarrow h_{b_{m,n}^{ori}} + c_{b_{m,n}^{ori}}$$

// Smooth h

Apply six times a circular convolution with filter $[1,1,1]/3$ to h.

//Extract the reference orientations

for $1 \leq k \leq n_{bins}$ **do**

if $h_k > h_{k^-}, h_k > h_{k^+}$ and $h_k \geq t \max(h)$ **then**

//Compute the reference orientation θ_{key}

$$\theta_{key} = \theta_k + \frac{\pi}{n_{bins}} \left(\frac{h_{k^-} - h_{k^+}}{h_{k^-} - 2h_k + h_{k^+}} \right)$$

➤ **Amélioration de la précision par interpolation des coordonnées [24]**

Visant à augmenter de façon significative la stabilité et la qualité de la mise en correspondance, cette étape s'effectue dans l'espace des échelles à trois dimensions, où $D(x, y, \sigma)$, qui n'est connu que pour des valeurs discrètes de x , y et σ , doit être interpolé. Cette interpolation s'obtient par un développement de Taylor à l'ordre 2 de la DoG $D(x, y, \sigma)$, en prenant comme origine les coordonnées du point-clé candidat. Ce développement s'écrit comme suit :

$$D(X) = D + \frac{\partial D^T}{\partial x} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \quad (II.4)$$

Où D et ses dérivées sont évalués au point-clé candidat et où $X = (x, y, \sigma)^T$ est un delta par rapport à ce point. Les dérivées sont estimées par différences finies à partir des points voisins connus de façon exacte. La position précise de l'extremum \hat{X} est déterminée en résolvant l'équation annulant la dérivée de cette fonction par rapport à X . Le point extremum trouvé est de la forme :

$$\hat{X} = -\frac{\partial^2 D^{-1}}{\partial X^2} \frac{\partial D}{\partial X} \quad (II.5)$$

Un delta \hat{X} supérieur à 0,5 dans l'une des trois dimensions signifie que le point considéré est plus proche d'un des voisins dans une octave. Dans ce cas, le point-clé candidat est mis à

jour et l'interpolation est réalisée à partir des nouvelles coordonnées. Sinon, le delta est ajouté au point candidat initial qui gagne ainsi en précision. Un algorithme de reconvergence similaire a été proposé dans l'implémentation temps-réel basée sur les pyramides hybrides de Lindeberg et Bretzner [28].

- Algorithme de lissage de Lindeberg

Input : u input digital image of $M \times N$ pixels.

Output : v output digital image of $M \times N$ pixels.

Parameters : σ applied blur.

$0 \leq \gamma \leq \frac{1}{2}$ parameter defining Δ_γ^{disc} the Laplacian finite difference scheme.

Temporary : P number of Euler iterations.

δt , Euler step size.

$\Delta + v$, $\Delta \times v$, $\Delta_\gamma^{disc} v$, auxiliary discrete Laplacians.

// Euler method setting

$$P \leftarrow \left\lceil 8 \left(1 - \frac{\gamma}{2} \right) \sigma^2 \right\rceil$$

$$\delta t \leftarrow \frac{\sigma}{P}$$

//Initialization

$$v \leftarrow u$$

//Euler Method

for $p = 1, \dots, P$ **do**

//Compute discrete Laplacian

for $0 \leq k \leq M - 1$ and $0 \leq l \leq N - 1$ **do**

$$\Delta + v_{k,l} \leftarrow v_{k+1,l} + v_{k,l+1} + v_{k,l-1} - 4v_{k,l}$$

$$\Delta \times v_{k,l} \leftarrow \frac{1}{2} (v_{k+1,l+1} + v_{k+1,l-1} + v_{k-1,l+1} - v_{k-1,l-1}) - 2v_{k,l}$$

$$\Delta_{\gamma}^{disc} v_{k,l} \leftarrow (1-\gamma)\Delta + v_{k,l} + \gamma\Delta_x v_{k,l}$$

Note: The half-sample symmetric boundary condition used.

//Euler iteration formula

for $0 \leq k \leq M-1$ and $0 \leq l \leq N-1$ **do**

$$v_{k,l} \leftarrow v_{k,l} - \delta t \Delta_{\gamma}^{disc} v_{k,l}$$

➤ Élimination des points-clés de faible contraste

La valeur de aux coordonnées précises du point-clé peut être calculée à partir du développement de Taylor de cette fonction, et constitue donc un extremum local. Un seuillage absolu sur cette valeur permet d'éliminer les points instables, à faible contraste.

➤ Élimination des points situés sur les arêtes

Les points situés sur les arêtes (ou contours) doivent être éliminés car la fonction DoG γ prend des valeurs élevées, ce qui peut donner naissance à des extremums locaux instables, très sensibles au bruit : si l'image devait subir un changement numérique même imperceptible, de tels points-clés peuvent se retrouver déplacés ailleurs sur la même arête, ou même simplement disparaître. Un point candidat à éliminer, si l'on considère les deux directions principales à sa position, est caractérisé par le fait que sa courbure principale le long du contour sur lequel il est positionné est très élevée par rapport à sa courbure dans la direction orthogonale. La courbure principale est représentée par les valeurs propres de la matrice hessienne H :

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (II.6)$$

Les dérivées doivent être évaluées aux coordonnées du point d'intérêt (x, y, σ) dans l'espace des échelles. Les valeurs propres de H sont proportionnelles aux courbures principales de D , dont seul le rapport r est intéressant. La trace de H représente la somme de ces valeurs, le déterminant son produit [12]. Par conséquent, en adoptant un seuil r_{th} sur le ratio des courbures ($r_{th} = 10$ dans la méthode originale de Lowe), un point-clé candidat va être retenu, selon le critère adopté par Lowe, si :

$$R = \frac{tr(H)^2}{\det(H)} = \frac{(r+1)^2}{r} < \frac{(r_{th}+1)^2}{r_{th}} \quad (II.7)$$

La vérification de ce critère est rapide, ne nécessitant qu'une dizaine d'opérations flottantes seulement. Lorsque ce critère n'est pas vérifié, le point est considéré comme localisé le long d'une arête et il est par conséquent rejeté.

Cette étape est inspirée de la technique de détection de points d'intérêt par l'opérateur de Harris [16] ; pour le seuillage, une matrice hessienne est utilisée au lieu de la matrice des moments d'ordre 2.

➤ **Assignment d'orientation**

L'étape d'assignment d'orientation consiste à attribuer à chaque point-clé une ou plusieurs orientations déterminées localement sur l'image à partir de la direction des gradients dans un voisinage autour du point. Dans la mesure où les descripteurs sont calculés relativement à ces orientations, cette étape est essentielle pour garantir l'invariance de ceux-ci à la rotation : les mêmes descripteurs doivent pouvoir être obtenus à partir d'une même image, quelle qu'en soit l'orientation [13]. Pour un point-clé donné (x, y, σ) , le calcul s'effectue sur $L(x, y, \sigma)$ à savoir le gradient de la pyramide dont le paramètre est le plus proche du facteur d'échelle du point. De cette façon, le calcul est également invariant à l'échelle. A chaque position dans un voisinage du point-clé, on estime le gradient par différences finies symétriques, puis son amplitude (c.-à-d. sa norme) $m(x, y)$, et son orientation $\theta(x, y)$:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (\text{II.8})$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (\text{II.9})$$

Un histogramme des orientations sur le voisinage est réalisé avec 36 intervalles, couvrant chacun 10 degrés d'angle [43]. L'histogramme est doublement pondéré : d'une part, par une fenêtre circulaire gaussienne de paramètre égal à 1,5 fois le facteur d'échelle du point-clé σ ; d'autre part, par l'amplitude de chaque point. Les pics dans cet histogramme correspondent aux orientations dominantes (Voir Figure.II.4). 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. A l'issue de cette étape, un point-clé est donc défini par quatre paramètres (x, y, σ, θ) . Il est à noter qu'il est parfaitement possible qu'il y ait sur une même image plusieurs points-clés qui ne diffèrent que par un seul de ces quatre paramètres (le facteur d'échelle ou l'orientation, par exemple).

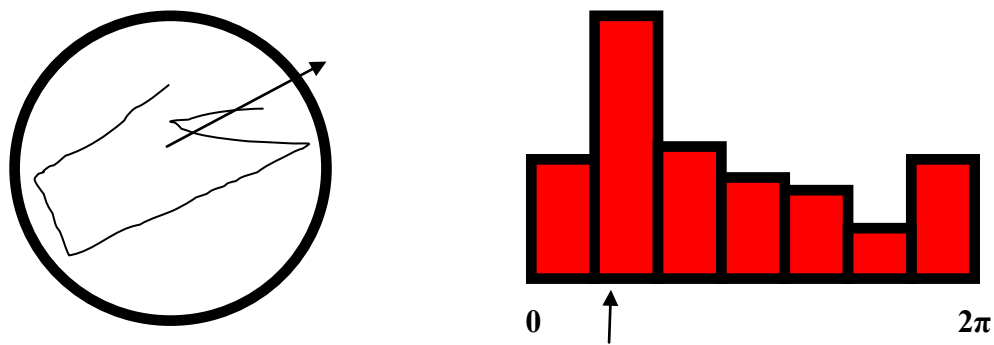


Figure II.4 Construction de l'histogramme des orientations.

➤ Descripteur de point-clé

Une fois les points-clés, associés à des facteurs d'échelles et à des orientations, détectés et leur invariance aux changements d'échelles et aux rotations assurée, arrive l'étape de calcul des vecteurs descripteurs, traduisant numériquement chacun de ces points-clés. À cette occasion, des traitements supplémentaires vont permettre d'assurer un surcroît de pouvoir discriminant en rendant les descripteurs invariants à d'autres transformations telles que la luminosité, le changement de point de vue 3D, etc. Cette étape est réalisée sur l'image lissée avec le paramètre de facteur d'échelle le plus proche de celui du point-clé considéré. Autour de ce point, on commence par modifier le système de coordonnées local pour garantir l'invariance à la rotation, 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 la magnitude du gradient sont calculés comme précédemment. L'orientation détermine l'intervalle à incrémenter dans l'histogramme, ce qui se fait avec un double pondération – par l'amplitude et par une fenêtre gaussienne centrée sur le point clé, de paramètre égal à 1,5 fois le facteur d'échelle du point-clé. 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 supérieures à 0,2 sont remplacées par 0,2 et l'histogramme est de nouveau normalisé, pour finalement fournir le descripteur SIFT du point-clé, de dimension 128.

Cette dimension peut paraître bien élevée, mais la plupart des descripteurs de dimension inférieure proposés dans la littérature présentent de moins bonnes performances dans les tâches de mise en correspondance pour un gain en coût de calculs bien modéré pour trouver le plus proche voisin [24].

➤ Utilisation pour la recherche d'objets dans des images

La problématique de base pour laquelle la méthode SIFT a été conçue est la suivante : peut-on trouver dans une image donnée (dite *image question* ou *image suspecte*), des objets déjà présents dans une collection d'images de référence préétablie. Dans la méthode originale de David Lowe, les points-clés et les descripteurs SIFT sont tout d'abord extraits des images de référence et stockés dans une sorte de base de données. Un objet est identifié dans l'image question en effectuant une comparaison de ses descripteurs à ceux des images de référence disponibles en base de données, fondée simplement sur la distance euclidienne. Parmi toutes les correspondances ainsi établies, des sous-ensembles (*clusters*) sont identifiés, au sein desquels la mise en correspondance est cohérente du point de vue des positions des points, des facteurs d'échelle et des orientations. Les clusters contenant au moins trois correspondances ponctuelles sont conservés. Dans chacun d'eux, on modélise la transformation permettant de passer de l'image question à l'image de référence, et on élimine les correspondances aberrantes par simple vérification de ce modèle. Enfin, Lowe applique un modèle probabiliste pour confirmer que la détection d'une correspondance d'objets entre l'image question et l'une des images de référence n'est pas due au hasard, baser sur l'idée que si de nombreux points n'ont pas pu être mis en correspondance c'est que l'on a peut-être affaire à un faux positif.

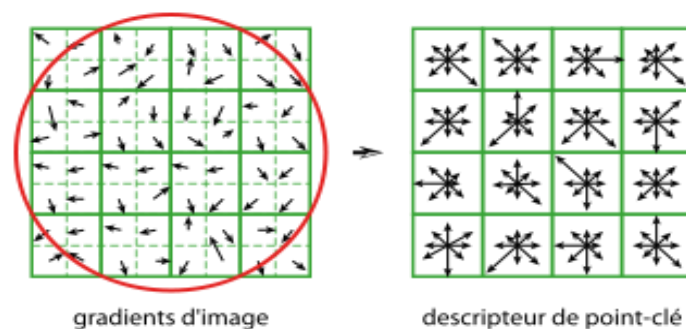


Figure II.5 Construction d'un descripteur SIFT.

- Construction de descripteur de point clé

Inputs: - $L_D = \{(O_{key}, S_{key}, X_{key}, Y_{key}, \mathcal{G}_{key})\}$ list of keypoints.

- $(\mathcal{G}_m v_{2,m,n}^o)$, scale-space gradient along x .

- $(\mathcal{G}_n v_{2,m,n}^o)$, scale-space gradient along y (see Algorithm 14).

Output: - $L_E = \{(O_{key}, S_{key}, X_{key}, Y_{key}, \mathcal{G}_{key}, \mathbf{F})\}$ list of keypoints with feature vector \mathbf{F} .

Parametres: - n_{hist} . The descriptor is an array of $n_{hist} \times n_{hist}$ orientation histograms.

- n_{ort} , number of bins in the orientation histograms.

Feature vectors \mathbf{f} have a length of $n_{hist} \times n_{hist} \times n_{ort}$

- λ_{descr} .

The Gaussian window has a standard deviation of $\lambda_{descr} \sigma_{key}$.

The patch ρ^{descr} is $2\lambda_{descr} \frac{n_{hist} + 1}{n_{hist}} \sigma_{key}$ wide.

Temporary: h_k^{ij} , array of orientation weighted histograms, $(i, j) \in \{1, \dots, n_{hist}\}$ and $k \in \{1, \dots, n_{ort}\}$

for each keypoint $\{(O_{key}, S_{key}, X_{key}, Y_{key}, \mathcal{G}_{key})\}$ **in do**

//Check if the keypoint is distant enough from the image borders

if $\sqrt{2}\lambda_{descr} \sigma \leq x_{key} \leq h - \sqrt{2}\lambda_{descr} \sigma$ and $\sqrt{2}\lambda_{descr} \sigma \leq y_{key} \leq w - \sqrt{2}\lambda_{descr} \sigma$ **then**

// Initialize the array of weighted histograms

for $1 \leq i \leq n_{hist}$, $1 \leq j \leq n_{hist}$ **and** $1 \leq k \leq n_{ort}$ **do** $h_k^{ij} \leftarrow 0$

// Accumulate samples of normalized patch ρ^{descr} in the array histograms (eq.(3.25))

for $m = \left[\left(x_{key} - \sqrt{2}\lambda_{desc} \sigma_{key} \frac{n_{hist} + 1}{n_{hist}} \right) / \delta_0 \right], \dots, \left[\left(x_{key} - \sqrt{2}\lambda_{desc} \sigma_{key} \frac{n_{hist} + 1}{n_{hist}} \right) / \delta_0 \right]$ **do**

for $n = \left[\left(y_{key} - \sqrt{2}\lambda_{desc} \sigma_{key} \frac{n_{hist} + 1}{n_{hist}} \right) / \delta_0 \right], \dots, \left[\left(y_{key} - \sqrt{2}\lambda_{desc} \sigma_{key} \frac{n_{hist} + 1}{n_{hist}} \right) / \delta_0 \right]$ **do**

// Compute normalized coordinates (eq.(3.24)).

$$\hat{x}_{m,n} = ((m\delta_{0_{key}} - x_{key}) \cos \mathcal{G}_{key} + (n\delta_{0_{key}} - y_{key}) \sin \mathcal{G}_{key}) / \sigma_{key}$$

$$\hat{y}_{m,n} = (-(m\delta_{0_{key}} - x_{key}) \sin \mathcal{G}_{key} + (n\delta_{0_{key}} - y_{key}) \cos \mathcal{G}_{key}) / \sigma_{key}$$

// Verify if the sample (m,n) is inside the normalized patch

ρ^{descr}

if $\max(|\hat{x}_{m,n}|, |\hat{y}_{m,n}|) < \lambda_{desc} \frac{n_{hist} + 1}{n_{hist}}$ then

// Compute normalized gradient orientation.

$$\mathcal{G}_{m,n} = \arctan 2\left(\mathcal{G}_m v_{\delta_{key}}^{o_{key}}, m, n, \mathcal{G}_n v_{\delta_{key}}^{o_{key}}, m, n\right) - \mathcal{G}_{key} \bmod 2\pi$$

// Compute the total contribution of the sample (m,n)

$$c_{m,n}^{descr} = e^{-\frac{\left\| (m\delta^{o_{key}}, n\delta^{o_{key}}) - (x_{key}, y_{key}) \right\|^2}{2(\lambda_{desc} \sigma_{key})^2}} \left\| \left(\mathcal{G}_m v_{\delta_{key}}^{o_{key}}, m, n, \mathcal{G}_n v_{\delta_{key}}^{\sigma_{key}}, m, n \right) \right\|$$

//Update the nearest histograms and the nearest bins

for $(i, j) \in \{1, \dots, n_{hset}\}^2$ such that $|\hat{x}^i - \hat{x}_{m,n}| \leq \frac{2\lambda_{hset}}{\pi_{hset}}$ and

$|\hat{y}^j - \hat{y}_{m,n}| \leq \frac{2\lambda_{hset}}{\pi_{hset}}$ do

for $k \in \{1, \dots, n_{hset}\}^2$ such that $|\theta^k - \theta_{m,n} \bmod 2\pi| < \frac{2\pi}{n_{ori}}$ do

$$h_k^{i,j} \leftarrow h_k^{i,j} + \left(1 - \frac{n_{hset}}{2\lambda_{d \min}} |\hat{x}_{m,n} - \hat{x}^i|\right) \left(1 - \frac{n_{hset}}{2\lambda_{d \min}} |\hat{y}_{m,n} - \hat{y}^j|\right) \left(1 - \frac{n_{ori}}{2\pi} |\theta_{m,n} - \theta^k \bmod 2\pi|\right)$$

//Build the feature vector \mathbf{f} from the array of weighted histograms

for $1 \leq i \leq n_{hist}, 1 \leq j \leq n_{hist}$ and $1 \leq k \leq n_{ori}$ do

$$f((i-1)n_{hist}n_{ori} + (j-1)n_{ori} + k) = h_k^{i,j}$$

for $1 \leq l \leq n_{hist} \times n_{hist} \times n_{ori}$ do

$$f_l \leftarrow \min(f_l, 0.2 \|f\|) / \text{*normalize and threshold } f \text{*}$$

Compute the 12 norm $f_i \leftarrow \min(\lfloor 512 f_i / \|f\| \rfloor), 255$ / * quantize to 8 bits integers

Added $(x, y, \sigma, \theta, f)$ to l_c

➤ Indexation des descripteurs et recherche de correspondances

L'indexation est l'opération de stockage des descripteurs SIFT des images de référence, d'une manière qui facilite l'identification des descripteurs correspondants de l'image

question. Lowe utilise un arbre kd pour indexer les descripteurs puis une méthode de recherche dans cet arbre modifiée par rapport à l'approche classique, appelée *Best bin first*. Cette dernière est capable de trouver les plus proches voisins d'un descripteur question avec une bonne probabilité de façon très économe en temps de calcul. Cet algorithme se fonde sur deux astuces. Tout d'abord, le nombre de boîtes (feuilles de l'arbre kd) à explorer pour trouver le plus proche voisin d'un descripteur question donné est limité à une valeur maximale fixée, par exemple à 200. Ensuite, les nœuds de l'arbre kd sont explorés dans l'ordre de leur distance au descripteur question, grâce à l'utilisation d'une file de priorité basée sur un tas binaire. [14] Par distance d'un nœud à un descripteur, on entend distance euclidienne de la boîte englobant des feuilles sous-jacentes à ce descripteur. Dans la plupart des cas, ce procédé fournit la meilleure correspondance, et, dans les cas restants, un descripteur très proche de celle-ci. De façon à déconsidérer les descripteurs faiblement discriminants (parce qu'ils sont trop souvent présents dans la base de référence, comme par exemple ceux qui sont extraits de motifs d'arrière-plan souvent répétés dans les images), Lowe recherche à la fois le plus proche voisin et le second plus proche voisin de chaque descripteur question. Lorsque le rapport des distances est supérieur à 0,8, la correspondance est éliminée, car considérée comme ambiguë (« bruit »). Par ce critère, Lowe parvient à éliminer 90 % des fausses correspondances en perdant moins de 5 % de correspondances correctes. Nous retrouvons plusieurs applications industrielles des SIFT :

- Assistance des conducteurs de voitures
- La photographie digitale.
- Le suivi des visages.
- Les films et les vidéos.
- Les jeux et la reconnaissance des gestes.
- Industrie électronique.
- Reconnaissance d'objets pour les engins mobiles.

II.1.1 Détection par SURF

Inspiré par SIFT et Bay. A proposé le détecteur SURF et le descripteur. Les caractéristiques de SURF présentent de meilleurs résultats en matière de répétabilité, de caractère distinctif et de robustesse, mais en même temps peut être calculé beaucoup plus rapidement grâce à l'utilisation de l'image intégrale [15], ce qui signifie que les dérivés gaussiens à différents niveaux d'échelle peuvent être approchés par des filtres de boîtes simples sans calculer l'espace global de l'échelle gaussienne. Similaire à SIFT, une grille rectangulaire de 4×4 sous-régions est définie (selon l'orientation principale) et une somme de réponses d'ondelettes de Haar (pondérée par un centre gaussien au point central d'intérêt) est calculée par région.

Le détecteur et le descripteur de SURF ne sont pas seulement plus rapides, mais le premier est également plus répandu et le dernier plus distinctif.

Le détecteur de points d'intérêt le plus utilisé est probablement le détecteur d'angle Harris [16], proposé en 1988, en fonction des valeurs propres de la matrice du second moment.

Cependant, les angles Harris ne sont pas invariants à l'échelle. Lindeberg A introduit le concept de la sélection automatique de l'échelle [17]. Cela permet de Détecter des points d'intérêt dans une image, chacune avec leur propre échelle caractéristique. Il a expérimenté à la fois le déterminant de la matrice hessienne ainsi que le Laplacien (qui correspond à la trace de la matrice hessienne) pour détecter Bloblike Structures. Mikolajczyk et Schmid ont raffiné cette méthode, créant une robustesse Et des détecteurs de caractéristiques invariables avec une grande répétabilité, qu'ils ont inventé Harris-Laplace et Hessian-Laplace [18]. Le descripteur, d'autre part, décrit une distribution des réponses Haar-ondelettes dans le quartier des points d'intérêt. Encore une fois, nous exploitons des images intégrales pour la vitesse.

Nous présentons également une nouvelle étape d'indexation basée sur le signe du Laplacien, qui augmente non seulement la vitesse de correspondance, mais aussi la robustesse du descripteur.

Afin de rendre le document plus autonome, nous discutons succinctement le concept d'images intégrales, tel que défini par [19]. Ils permettent la mise en place rapide de filtres de convolution de type caisse. L'entrée d'une image intégrale $I_{\Sigma}(x)$ à un emplacement $X = (x, y)$ représente la somme de tous les pixels dans l'image d'entrée I d'une région rectangulaire formée par le point x et l'origine, $I_{\Sigma}(X) = \sum_{i=0}^{i<x} \sum_{j=0}^{j<y} I(i, j)$.

Avec I_{Σ} calculé, il ne faut que quatre ajouts pour calculer la somme des intensités sur n'importe quelle surface rectangulaire droite, indépendamment de sa taille.

II.1.1.1 Détecteur Fast-Hessian

Nous basons notre détecteur sur la matrice Hessian en raison de sa bonne performance en temps de calcul et précision. Cependant, plutôt que d'utiliser une mesure différente pour sélectionner l'emplacement et l'échelle (comme cela a été fait dans le détecteur Hessian-Laplace), nous comptons sur le déterminant du Hessian pour les deux. Compte tenu d'un point $x = (x, y)$ dans une image I , la matrice Hessienne $H(x, \sigma)$ en x à l'échelle σ est définie comme suit :

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (\text{II.10})$$

Où $L_{xx}(x, \sigma)$ est la convolution de la dérivée gaussienne de second ordre $\frac{\partial^2}{\partial x^2} g(\sigma)$

Avec l'image I dans le point x , et de manière similaire pour $L_{xy}(x, \sigma)$ et $L_{yy}(x, \sigma)$.

Les Gaussiens sont optimaux pour l'analyse de l'espace échelle [18].

Les filtres à boîte 9×9 de la Figure I.12 sont des approximations pour les dérivés gaussiens de second ordre avec $\sigma = 1.2$ et représentent notre échelle la plus basse (c'est-à-dire la résolution spatiale la plus élevée). Nous désignons nos approximations par D_{xx} , D_{yy} et D_{xy} .

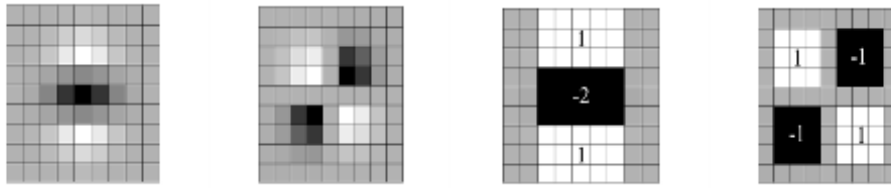


Figure II.6 De gauche à droite: les dérivées partielles gaussiennes (discrétisées et recadrées) en direction de y et en direction de xy , et nos approximations en utilisant des filtres à caisse. Les régions grises sont égales à zéro.

Les poids Appliqué aux régions rectangulaires est simple pour l'efficacité de calcul, mais nous devons équilibrer davantage les poids relatifs dans l'expression pour le déterminant de Hessian avec :

$$\frac{|L_{xy}(1.2)|_F |D_{xx}(9)|_F}{|L_{xx}(1.2)|_F |D_{xx}(9)|_F} = 0.912... \approx 0.9, \text{ où } |x|_F \text{ est La norme de Frobenius. Cela produit}$$

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.912D_{xy})^2 \quad (\text{II.11})$$

En outre, les réponses des filtres sont normalisées par rapport à la taille du masque, cela garantit une norme constante de Frobenius pour toute taille de filtre.

II.1.1.2 Le descripteur SURF

La bonne performance de SIFT par rapport à d'autres descripteurs [19] est remarquable.

Le mélange d'informations grossièrement localisées et la répartition des fonctionnalités liées au dégradé semblent produire un bon pouvoir distinctif tout en réduisant les effets des erreurs de localisation en termes d'échelle ou d'espace. Utiliser des forces relatives et les orientations des gradients réduisent l'effet des changements photométriques.

Le descripteur SURF proposé est basé sur des propriétés similaires, avec une complexité réduite encore plus loin. La première étape consiste à fixer une orientation reproductible à partir d'informations provenant d'une région circulaire autour du point d'intérêt.

Ensuite, nous construisons une région carrée alignée sur l'orientation sélectionnée et extrayons le descripteur SURF. Ces deux étapes sont expliquées à leur tour.

En outre, nous proposons également une version verticale de notre descripteur (U-SURF) qui n'est pas invariante à la rotation de l'image et donc plus rapide à calculer et mieux adapté aux applications où la caméra reste plus ou moins horizontale.

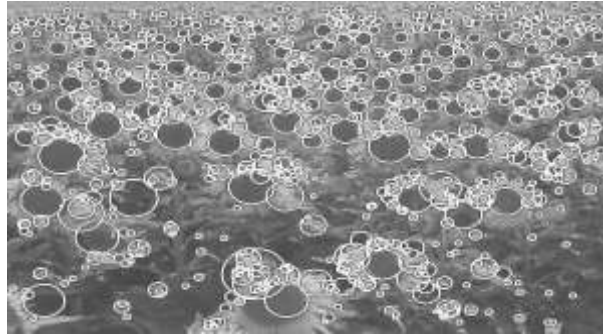


Figure II.7 Points d'intérêt détectés pour un champ de tournesol.

➤ Affectation d'orientation

Pour être invariant à la rotation, nous identifions une orientation reproductible pour les points d'intérêt. Pour cela, nous calculons d'abord les réponses Haar-ondelettes dans la direction x et y , représentées sur la Figure I.13, et ceci dans un quartier circulaire de rayon $6s$ autour du point d'intérêt, avec s l'échelle à laquelle le point d'intérêt a été détecté. De plus, l'étape d'échantillonnage dépend de l'échelle et choisit d'être s . Conformément au reste, les réponses des ondelettes sont calculées à cette échelle actuelle s . En conséquence, à grande échelle, la taille des ondelettes est grande.

II.1.1.3 Les limitations du SIFT et SURF

➤ Évaluation des performances

Différentes évaluations ont été menées pour comparer les performances des ces détecteurs et descripteurs. Les SURF extraient généralement moins de points que les SIFT n'extraient de clés ce qui peut présenter un avantage ou un inconvénient selon l'application. Dans certains cas en effet, seules quelques associations de points sont nécessaires et les SURF sont tout à fait adaptés. A l'inverse, d'autres cas nécessitent l'extraction de nombreuses associations (reconstruction 3D par exemple) auquel cas les SIFT seront un meilleur choix [17]. Notons que dans l'idéal, une combinaison des deux descripteurs locaux devrait être optimale du fait de la complémentarité de leur localisation (sur les coins pour les clés SIFT et plutôt sur des régions caractéristiques pour les SURF). Les performances en termes de pourcentage de bons appariements des SURF sont généralement très proches de celles des SIFT pour un coût calculatoire bien moindre. En outre, plusieurs applications (localisation par imagerie, reconnaissance faciale, ...) ont montré la supériorité des SURF sur les SIFT [8] [9] notamment en termes de robustesse aux variations des conditions d'illumination. La conclusion et les recommandations pour les travaux de fonctionnalité dans ce sujet conduisent à l'amélioration de ces algorithmes puissants pour atteindre un niveau d'efficacité supérieur dans le cadre de sa performance.

II.1.2 Détection par algorithme AKAZE

Dans ce chapitre, nous présentons AKAZE features [34], Une nouvelle fonctionnalité 2D multi scalaire de détection et algorithme de description dans des espaces d'échelle non linéaire. Les méthodes précédentes telles que SIFT ou SURF trouvent des caractéristiques dans l'espace de l'échelle gaussienne. Les approches antérieures détectent et décrivent les caractéristiques à différents niveaux d'échelle en construisant ou en rapprochant l'espace de l'échelle gaussienne d'une image. Cependant, le flou gaussien ne respecte pas les limites naturelles des objets et lisse le même degré de précision et de réduction, réduisant l'exactitude de la précision et la neutralité. En revanche, nous détectons et décrivons les fonctionnalités 2D dans un espace à échelle non linéaire au moyen d'un filtrage non linéaire. Dans ce cas, On peut brancher s'adapte localement aux données d'image, réduisant le bruit mais En conservant les limites des objets, en obtenant une précision supérieure de la précision et une influence indirecte.

II.1.3 Descripteur KAZE

II.1.3.1 Introduction

Le traitement d'image à plusieurs niveaux est un outil très important dans les applications de vision par ordinateur. Nous pouvons résumer une image en détectant automatiquement les caractéristiques d'intérêt à différents niveaux d'échelles. Les données détectées peuvent être obtenues à l'aide d'une description géographique unique. Ces algorithmes de fonctionnalité multi scalaire sont un composant clé dans les systèmes d'imagerie informatiques modernes, tels qu'ils comprennent [24], la classification visuelle [25] et la structure 3D à grande échelle de Motion (SfM) [26]. L'idée principale des méthodes multi-scalaires est assez simple: Créez l'espace de l'échelle d'une image en filtrant l'image originale avec une fonction appropriée au fur et à mesure de l'augmentation du temps ou de l'échelle. Dans le cadre de l'espace de vente des Russes, ceci a permis de retrouver l'image originale avec un écart type croissant de Gaussien kernel of. Pour les valeurs de plus grande taille, nous obtenons des représentations d'image plus simples. Avec une représentation d'image multi-scalaire, nous pouvons détecter et décrire les caractéristiques d'image à différents niveaux d'échelle ou des résolutions. Plusieurs auteurs [27,28] ont montré que, sous certaines hypothèses générales, le noyau gaussien et le décalage des éléments diversifiés sont des éléments de neutralisation de l'analyse de l'espace de base. Cependant, il est important de noter ici que l'espace à échelle gaussienne n'est qu'une instance de diffusion linéaire, car d'autres espaces à échelle linéaire sont également possibles [29].

Le noyau gaussien est probablement l'option la plus simple (mais pas le seul) pour créer une représentation spatiale d'une image. Cependant, il présente des inconvénients importants. Dans l'espace de l'échelle gaussienne, les avantages de la sélection des échelles plus grossières sont la réduction du bruit et l'accent sur une structure plus importante. La raison

en est que le flou gaussien ne respecte pas les limites naturelles des objets et lisse au même degré les détails et le bruit à tous les niveaux d'échelle. Cette perte de localisation augmente tant que nous détectons les caractéristiques à des niveaux d'échelle plus grossiers, où la quantité de flou gaussien est plus élevée.

Nos fonctions s'appellent KAZE, en hommage à Iijima [30], le père de l'analyse de l'espace d'échelle.

KAZE est un mot japonais qui signifie vent. Dans la nature, le vent est défini comme le flux d'air à grande échelle et normalement ce flux est régi par des processus non linéaires. Dans ce cas, nous faisons l'analogie avec des processus de diffusion non linéaire dans le domaine de l'image.

II.1.3.2 La description par AKAZE

Accelerated-KAZE Features utilise un nouveau cadre mathématique appelé Fast Explicit Diffusion (FED) intégré dans un cadre pyramidal pour accélérer considérablement le calcul de l'espace non linéaire. En outre, nous calculons un descripteur Binary Différentiel Local (MLDB) [8] robuste qui exploite des informations sur les gradients à partir de l'espace échelle non linéaire. A-KAZE [34] obtient des résultats comparables à KAZE dans certains ensembles de données.

Du point clé comme dans KAZE, et la grille de LDB tournée en conséquence. Au lieu d'utiliser la moyenne de tous les pixels à l'intérieur de chaque subdivision de la grille, nous sous-échantillons les grilles par étapes qui sont fonction de l'échelle σ de la fonctionnalité. L'échantillonnage dépendant de l'échelle fait tour à tour le descripteur.

II.1.3.3 Caractéristiques KAZE

Dans cette section, nous décrivons notre nouvelle méthode pour la détection et la description des fonctionnalités dans des espaces d'échelle non linéaire. Compte tenu d'une image d'entrée, nous construisons l'espace d'échelle non linéaire jusqu'à un temps d'évolution maximal en utilisant des techniques AOS (Additive Operator Splitting) (Fractionnement de l'opérateur additif) et une diffusion de conductance variable. Ensuite, nous détectons les caractéristiques 2D intéressantes qui présentent un maximum de la détermination normalisée par l'échelle de la limite de l'hessien dans le domaine des lignes secondaires [32]. En définitive, vérifiez l'orientation principale du point clé et obtenez un descripteur invariant à l'échelle et à la rotation en considérant les dérivées d'image de premier ordre. Maintenant, nous décrivons chacune des étapes principales de notre formulation.

II.1.4 Détection des features par AKAZE

➤ Détection des features

Nous calculons le déterminant pour le Hessien pour chacune des images de filtres L_i dans l'espace d'échelle non linéaire. L'ensemble de l'opérateur multi-échelle différentiel est normalisé par rapport à l'échelle, En utilisant un facteur d'échelle a normalisé qui prend en compte l'octave de chaque image particulière dans l'espace d'échelle non linéaire :

$$L_{Hessian}^i = \sigma_{i,norm}^2 (L_{xx}^i L_{yy}^i - L_{xy}^i L_{xy}^i) \quad (II.12)$$

Pour calculer les dérivées de second ordre, nous utilisons des filtres Scharr concaténés avec la taille σ_i , norme. Les fléchisseurs de Scharr rappellent l'invariance de rotation mieux que les autres filtres ou la différenciation des différences centrales [33]. Nous cherchons tout d'abord les maximums de la réponse du détecteur dans l'emplacement spatial. À chaque niveau d'évolution i , nous vérifions que la réponse du détecteur est supérieure à un seuil prédéfini et c'est un maximum dans une fenêtre de 3×3 pixels. Ceci est fait pour éliminer rapidement les réponses non-maximums. Ensuite, pour chacun des potentiels, nous vérifions que la réponse est un maximum par rapport à d'autres points clés du niveau $i+1$ et $i-1$, respectivement, directement sur le dessus et directement sur les pixels de taille $\times \sigma_i$. Enfin, la position 2D du point clé est estimée avec une précision de sous-pixel en fixant une fonction quadratique 2D au déterminant de la réponse hessienne dans un quartier de 3×3 pixels et en trouvant son maximum.

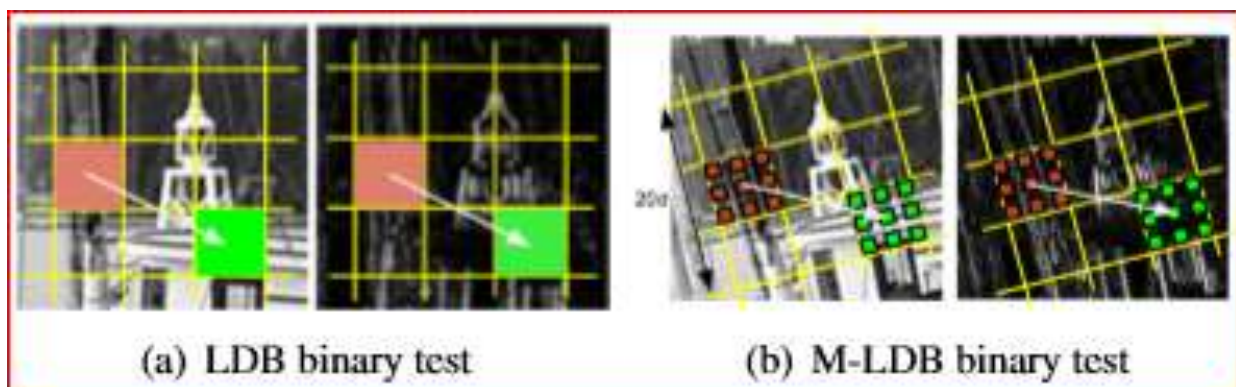


Figure II.8 LDB et les tests binaires M-LDB proposés entre les divisions de grille autour d'un point clé, montrés pour l'intensité et les gradients dans x . M-LDB comprend la rotation et le sous-échantillonnage qui dépend de l'échelle.

II.1.5 Détection de visage par descripteur AKAZE

➤ Détection de visage

Détection de visage a beaucoup d'applications dans la vie quotidienne. C'est une première étape dans n'importe quels systèmes de traitement de visage tels que reconnaissance de visage, vérification de visage. Il est également un problème principal qu'on doit surmonter dans le domaine d'interaction homme machine [35]. Les visages dans l'image sont détectés en balayant une fenêtre de taille 19x27 sur l'image d'entrée. À chaque endroit, celle-ci est déterminée comme un visage ou non grâce à la réponse du réseau de neurones. Pour que tous les visages aux tailles différentes puissent détecter, l'image d'entrée est traitée multi-échelle avec laquelle elle est considérée comme plusieurs images d'entrée qui ont des tailles [36].

II.2 Algorithme de détection

II.2.1 Algorithme AKAZE

Cette section fournit une brève introduction à la fonction AKAZE. Une analyse plus poussée démontre les avantages et les inconvénients d'AKAZE pour réaliser des performances en temps réel pour les solutions matérielles. Enfin, un descripteur binaire compatible avec le matériel est introduit pour réduire le coût de la mémoire et la complexité informatique.

II.2.2 Présentation d'AKAZE

La caractéristique AKAZE contient trois étapes principales: la construction pyramidale non linéaire, l'emplacement des points clés et la génération de descripteurs binaires. Dans la première étape, AKAZE emploie l'équation de Perona-Malik (P.M.) [37] pour construire une pyramide à échelle non linéaire. Pour construire différents sous-niveaux dans la pyramide, cette méthode diffuse l'image originale à une série de niveaux d'échelle croissants en utilisant l'équation (II.13), où I est la luminance de l'image et k le facteur de contraste. Afin d'accélérer le processus de diffusion, AKAZE adopte un schéma de diffusion explicite rapide (FED) [37, 38] qui se rapproche des solutions par des itérations. Chaque itération peut diffuser l'image avec une petite échelle. Basé sur des étapes d'échelle variables (plutôt que constantes), FED réduit considérablement le nombre d'itérations:

$$\frac{\partial I}{\partial t} = \text{div} \left(\frac{\nabla I}{1 + |\nabla I|^2 / k^2} \right) \quad (\text{II.13})$$

Les points clés sont situés dans la deuxième étape. Une fois la pyramide construite, le déterminant de la matrice Hessienne normalisée est calculé (Equation II.14). Ensuite, les

maxima locaux à chaque sous-choix sont sélectionnés comme points clés candidats. Afin de rechercher les extrêmes de l'échelle, un point potentiel est comparé à d'autres points clés candidats dans une fenêtre $\sigma \times \sigma$ du sous-niveau $i-1$ à $i + 1$:

$$H_{norm} = \sigma^2 \begin{vmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{vmatrix} \quad (II.14)$$

La troisième étape d'AKAZE introduit un descripteur binaire de différence local modifié (MLDB). Typiquement, la génération de descripteurs de MLDB est divisée en trois étapes: estimation d'orientation principale, rotation de patch d'échantillons et génération de descripteurs binaires. Tout d'abord, MLDB estime l'orientation principale en utilisant une méthode de type SURF basée sur l'histogramme. Ensuite, le motif d'échantillonnage est tourné en fonction de la direction du point clé. Enfin, les descripteurs binaires sont générés par comparaison entre les grilles dans trois canaux (une luminance et deux dérivations de premier ordre tournées).

II.3 Conclusion

Nous avons vu dans ce chapitre l'algorithme et technique de conception qui peuvent réaliser une accélération en temps réel pour l'extraction de fonctionnalités. Le système proposé met en relief les caractéristiques d'AKAZE dans une OSA d'architecture système très flexible.

Nous avons conclu les points de caractéristique sont localisés par un algorithme Accelerated-KAZE amélioré (AKAZE), à savoir les points de caractéristique détectés dans l'espace non linéaire construit par Fast Explicit Diffusion (FED) pour améliorer la vitesse de traitement, et la fonction de conductance variable.

**C
H
A
P
I
T
R
E**

III

**APPLICATION DE
DESCRIPTEUR AKAZE**

Chapitre III : Application de descripteur AKAZE

III.1 Introduction à la bibliothèque OpenCV

Depuis une dizaine d'années, une nouvelle branche de l'informatique est apparue : la Vision par Ordinateur (« Computer Vision » en anglais). Cette discipline consiste à regrouper les caractéristiques géométriques des images numériques (orientations, positions dans l'espace) afin de créer des modèles en 3 dimensions. On l'associe souvent à l'intelligence artificielle car ces algorithmes permettent de situer automatiquement des images tout en y identifiant certaines formes.

OpenCV est une bibliothèque informatique libre de droit (mise à part pour l'utilisation de SIFT et SURF) regroupant la majorité des algorithmes développés en vision par ordinateur. Bien connue dans le milieu de la recherche, cette bibliothèque est régulièrement mise à jour et garantie pour 3DReshaper une certaine stabilité dans les calculs. Même si ces performances sont très souvent citées dans la littérature spécialisée (photogrammétrie, vision par ordinateur ou traitement d'image), il est important de garder en tête que certains algorithmes sont perfectibles et que la dépendance en un système « gratuit » ne garantit pas la pérennité de la bibliothèque [39].

On retrouve dans OpenCV différents modules nécessaires dans différents projets. La compilation en C++ permet de créer des bibliothèques de fonctions utilisables ensuite dans l'environnement de 3DReshaper. Pour la calibration les modules «core», «calib3D», «features», «flann», «highgui» et «imgproc» ont été utilisés [40].

III.2 La bibliothèque QT

QT est une bibliothèque écrite en C++ et destinée à la création d'applications graphiques. Dans ce projet, cette bibliothèque est utilisée pour créer une fenêtre principale servant d'interface graphique entre l'utilisateur et l'application. Par exemple, l'application reçoit en entrée des informations de la part de l'utilisateur par des objets tels que les boutons, les curseurs et lui fournit des résultats en sortie à travers des objets tels que les labels contenant des images. La bibliothèque QT offre une variété de composants d'interface graphique facilitant de nombreuses opérations.

III.3 Les paramètres de descripteur AKAZE

Les paramètres de descripteur AKAZE::AKAZE utilisées sont :

- **descriptor_type** –Type de descripteur extrait : DESCRIPTOR_KAZE, DESCRIPTOR_KAZE_UPRIGHT , DESCRIPTOR_MLDB or DESCRIPTOR_MLDB_UPRIGHT.
- **descriptor_size** –Taille du descripteur en bits.0 -> Taille réelle
- **descriptor_channels** -Nombre de canaux dans le descripteur (1, 2, 3)
- **threshold** - Seuil de réponse du détecteur au point d'acceptation
- **octaves** - Evolution maximale d'octave de l'image
- **sublevels** - Nombre par défaut de sous-niveaux par niveau d'échelle
- **diffusivity** - Type de diffuseur. DIFF_PM_G1, DIFF_PM_G2, DIFF_WEICKERT ou DIFF_CHARBONNIER.

III.3.1 Traitement d'images

- Elle propose la plupart des opérations classiques en traitement bas niveau des images.
- lecture, écriture et affichage d'une image ;
- calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs ;
- lissage, filtrage ;
- seuillage d'image (seuillage adaptatif) ;
- segmentation (composantes connexes) ;
- morphologie mathématique.

III.3.2 Traitement des vidéos

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

- Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra)
- Détection de droites, de segment et de cercles par Transformée de Hough
- Détection de visages par la méthode de Viola et Jones
- Cascade de classifieurs boostés
- Détection de mouvement, historique du mouvement
- Détection de points d'intérêts
- Estimation de flux optique
- Enveloppe convexe
- Ajustement d'une ellipse à un ensemble de points par la méthode des moindres carrés.

III.3.3 Le moteur de production « CMake »

Le CMake est comparable au programme " Make " dans le sens où le processus de construction logicielle est entièrement contrôlé par des fichiers de configuration, appelés **CMakeLists.txt** dans le cas de CMake.

Mais CMake ne produit pas directement le logiciel final. Cela permet aux développeurs d'utiliser leur environnement de développement préféré comme à leur habitude. C'est cette utilisation des outils habituels de développement qui distingue CMake des autres systèmes de production.

Le nom « CMake » est l'abréviation de « Cross Platform Make ». Malgré l'utilisation de « Make » dans son nom, CMake est une application séparée et de plus haut niveau que l'outil Make.

La manière la plus simple d'utiliser OpenCV dans n'importe quel code est d'utiliser CMake.

Quelques avantages :

Pas besoin de modifier quoi que ce soit lors du transfert entre Linux et Windows.

Peut facilement être combiné avec d'autres outils par CMake (c.-à-d. Qt, ITK et VTK).

Pour créer un fichier CMakeLists.txt, ça devrait ressembler à ça :

```
cmake_minimum_required(VERSION 2.8)
project( DisplayImage )
find_package( OpenCV REQUIRED )
add_executable( DisplayImage DisplayImage.cpp )
target_link_libraries( DisplayImage ${OpenCV_LIBS} )
```

III.4 Application du descripteur AKAZE

Dans cette partie, nous avons appliqué, le descripteur AKAZE sur un ensemble des vidéos des personnalités publiques. Dans la première séquence vidéo, nous avons appliqué le descripteur AKAZE sur un seul visage vu de face avec un fond statique. Le contraste dans cette séquence est quasi stable et les artefacts sont très faibles. Dès la première image le descripteur AKAZE cherche à localiser les points invariants dans l'image, particulier le visage avec le nez, les deux yeux. Une partie des mains est inclus dans la détection comme une mesure en faux positive. Aussi la réflexion de la lumière sur le mur arrière montre qu'une partie du fond contient aussi des points invariants. Ces points sont considérés comme des faux négatives (Voir figure ci-dessous)



Figure III.1 Exemple d'une séquence vidéo dans un studio.



Figure III.2 Exemple d'une séquence vidéo sur une scène

Dans un deuxième exemple, nous considérons une séquence d'un visage avec des cheveux long et une peau plus claire avec des yeux fermes la plupart du temps. Le descripteur AKAZE tente de localiser les points invariants à savoir la bouche, le nez, les yeux et les sourcilles. Aussi une partie de l'oreille apparait comme une structure invariante dans la séquence vidéo. Le descripteur AKAZE réussit à localiser les points ou structures invariantes sur un visage lorsque les conditions d'éclairage sont stables (voir figure III.2).



Figure III.3 Exemple d'une séquence vidéo d'un film.

Dans un troisième exemple, nous considérons une séquence d'un visage qui bouge avec les mains. Le descripteur AKAZE réussit à localiser les points ou structures invariantes sur un visage et la main (voir figure III.3).



Figure III.4 Exemple d'une séquence vidéo en plein air.

Dans un quatrième exemple, nous considérons une séquence d'un ensemble de visages dans le plein air. Le descripteur AKAZE tente de localiser les points invariants à savoir l'ensemble partis de corps. Le descripteur AKAZE réussit à localiser les points ou structures invariantes sur un visage lorsque les conditions d'éclairage sont variant à cause des rayons de soleil (voir figure III.4).



Figure III.5 Exemple d'une séquence vidéo dans un lieu fermé.

Dans un cinquième exemple, nous considérons une séquence d'un visage avec des cheveux et une peau plus claire avec des objets sur le visage (lunette). Le descripteur AKAZE tente de localiser les points invariants à savoir la bouche, le nez, les yeux avec un fond gris. Le descripteur AKAZE réussit à localiser les points ou structures invariantes sur un visage lorsque les conditions d'éclairage sont stables (voir figure III.5).



Figure III.6 Exemple d'observation en full.

Dans un sixième exemple, nous considérons une séquence d'un ensemble de visages dans le plein air. Le descripteur AKAZE tente de localiser les points invariants à savoir l'ensemble partis de corps. Le descripteur AKAZE réussit à localiser les points ou structures invariants sur un visage lorsque les conditions d'éclairage sont variées à cause des rayons de soleil (voir figure III.6).

III.5 La construction du descripteur AKAZE

```
//SURF
//cv::SurfFeatureDetector detector(1000);
//cv::SurfDescriptorExtractor extractor;
Ptr<Feature2D> f2d = xfeatures2d::SIFT::create(nfeatures, nOctaveLayers, contrastThreshold);
//Ptr<Feature2D> f2d = xfeatures2d::BoostDesc::create();

//Ptr<Feature2D> f2d = FeatureDetector::create("AKAZE");
```

La création d'un pointeur intelligent de type `<Feature2D>` avec des propriétés partagées, et le passage d'un objet `xfeatures2d::SIFT` afin d'accéder aux paramètres d'entrée du descripteur.

```
// Ptr<Feature2D> f2d = xfeatures2d::SURF::create();
// Ptr<DescriptorMatcher> matcher = makePtr<BFMatcher>(NORM_L2);
// std::vector< cv::DMatch > matches(cv::NORM_L2);
cv::BFMatcher matcher;
std::vector< cv::DMatch > matches(cv::NORM_L2);
```

Ici `DMatch` est une classe pour les descripteurs de points clés correspondants à un : indice de descripteur de requête, indice de descripteur de train, index d'images de tarin et distance entre descripteurs.

```
namedWindow("CLASSIC AKAZE FLOW in OpenCV ", 1);
```

La fonction `namedWindow` crée une fenêtre qui peut être utilisée comme espace réservée pour les images et les barres. Les fenêtres créées sont mentionnées par leurs noms.

```
cap >> frame;
cvtColor(frame, gray, CV_BGR2GRAY);
cap >> frame;
cvtColor(frame, gray, CV_BGR2GRAY);
normalize(gray, gray, 0, 255, NORM_MINMAX);
cap >> frame;
cvtColor(frame, gray, CV_BGR2GRAY);
```

Cette commande est une classe qui représente le passage de format de couleur `BGR` (Blue Green Red) à l'échelle de gris (`GRAY`).

III.6 Discussion des résultats

Parmi les caractéristiques que nous avons extraites, la performance de l'AKAZE est testée à l'aide de la répétabilité du détecteur et des mesures de précision correspondantes.

Le paramètre pyramide est sélectionné en raison de la flexibilité avec des octaves résultant de l'architecture proposée. Il y a deux octaves, chacune avec quatre sous-niveaux. La précision du point fixe est adoptée dans l'AKAZE avec différentes largeurs de bits dans les différentes étapes. Les résultats montrent que l'algorithme obtient une performance similaire par rapport à l'algorithme AKAZE d'origine. Après l'extraction des fonctionnalités de KAZE, nous appliquons deux approches pour représenter chaque clip vidéo, qui est un sac de mots spatial et un codage spatial peu rigoureux.

III.7 Conclusion

Dans notre travail d'extraction, nous avons extrait des caractéristiques de la fonction A-KAZE présente d'excellente performance en matière de détection et de description, tout en disposant d'exigences minimales de stockage de calcul et de descripteurs. Le code des fonctionnalités A-KAZE est implémenté en C++ en utilisant certaines fonctionnalités de la bibliothèque OpenCV.

Le système a de meilleures performances et robustesse en temps réel après une amélioration partielle de l'algorithme du système. C'est une plate-forme numérique qui montre que l'application de l'environnement extérieur changeant. En raison de la capacité partielle limitée des caractéristiques décrivant. Le patch moyen qui combine l'apprentissage en profondeur sera une recherche chaude dans le futur.

CONCLUSION GENERALE

CONCLUSION GÉNÉRALE

Nous avons abordé dans ce travail la problématique de détection de visages selon des représentations locales et globales. Nous avons présenté un système complet de détection de visages à partir de descripteurs locaux, dont nous avons étudié les différentes étapes. L'objectif des descripteurs locaux est de fournir une description invariante aux changements pouvant survenir entre deux images représentant le même visage ou la même scène. La description doit cependant rester suffisamment discriminante pour que la similarité soit suffisamment significative.

REFERENCES
BIBLIOGRAPHIQUES

Bibliographie

- [1] : Yann, Le Cun sur l'apprentissage prédictif, 2016.
- [2] : King-Sun FU, "Syntactic Methods in Pattern Recognition". Academic Press, 1974.
- [3] : H. B. Barlow. Unsupervised Learning. Neural Computation, vol. 1, no. 3, pages 295-311, 1989.
- [4] : L. P. Kaelbling, M. Littman and A. Moore. Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research, vol. 4, issue 1, pages 237-285, 1996.
- [5] : O. Chapelle, B. Schölkopf and A. Zien. *Semi-supervised learning*. Adaptive computation and machine learning. MIT Press, 2006.
- [6] : Ming-Hsuan Yang, David J. Kriegman et Narendra Ahuja, "Detecting Faces in Images: A Survey", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol 24, No 1, pp 33-58, January 2002
- [7] : H. Rowley, S. Baluja et T. Kanade, "Neural Network based Face Detection", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp 23-38, 1996
- [8] : S.CANU. Machines à noyaux pour l'apprentissage statique. Techniques de l'ingénieur-Dossier : TE5255. 2007.
- [9] : Pablo F. Alcantarilla, Adrien Bartoli, and Andrew J. Davison, KAZE Features, Université d'Auvergne, Clermont Ferrand, France, 2012
- [10] : Herbert Bay, Andreas Ess, Tinne Tuytelaars et Luc Van Gool, « *SURF: Speeded Up Robust Features* », *Computer Vision and Image Understanding*, vol. 110, n° 3, p. 346-359, 2008.
- [11] : D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision (IJCV)*, vi, 17, 57, 58, 61, 2004.
- [12] : C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference*, 1988.
- [13] : C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(5), 1997.
- [14] : S. Edelman, N. Intrator, and T. Poggio. Complex cells and object recognition, 1997.
- [15] : Baumberg, A.: Reliable feature matching across widely separated views. In: *CVPR*, (2000).
- [16] : Harris, C., Stephens, M.: A combined corner and edge detector. In: *Proceedings of the Alvey Vision Conference*, (1988).

- [17] : Lindeberg, T.: Feature detection with automatic scale selection. IJCV 30(2) (1998).
- [18] : Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: ICCV. Volume 1, (2001).
- [19] : Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (1), (2001).
- [20] : Koenderink, J.: The structure of images. Biological Cybernetics 50 (1984).
- [21] : Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI 27 (2005).
- [22] : VALGREN C., LILIENTHAL A., « SIFT, SURF and Seasons: Long-Term Outdoor Localization Using Local Features », 3rd European Conference on Mobile Robots, Freiburg, Germany, 2007.
- [23] : DREUW P., STEINGRUBE P., HANSELMANN H., NEY H., « SURF-Face: Face Recognition Under Viewpoint Consistency Constraints », British Machine Vision Conference (BMVC), London , 7-10 September, 2009.
- [24]: Liu, C., Yuen, J., Torralba, A.: Dense scene alignment using SIFT flow for object recognition. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2009)
- [25]: Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2006)
- [26]: Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a Day. In: Intl. Conf. on Computer Vision (ICCV), Kyoto, Japan (2009)
- [27]: Koenderink, J.: The structure of images. Biological Cybernetics 50(1984) 14 Pablo F. Alcantarilla, Adrien Bartoli, and Andrew J. Davison
- [28]: Lindeberg, T.: Feature detection with automatic scale selection. Intl. J. of Computer Vision 30(1998)
- [29]: Duits, R., Florack, L., J. De Graaf, J., ter Haar Romeny, B.: On the axioms of scale space theory. Journal of Mathematical Imaging and Vision 20(2004)
- [30]: Weickert, J., Ishikawa, S., Imiya, A.: Linear scale-space has first been proposed in Japan. Journal of Mathematical Imaging and Vision 10(1999)
- [31]: X. Yang and K. T. Cheng. LDB: An ultra-fast feature for scalable augmented reality. In IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR), 2012
- [32] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. Computer Vision and Image Understanding 110, 346–359 (2008)
- [33]: J. Weickert and H. Schar. A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. Journal of Visual Communication and Image Representation, 13:103–118, 2002.
- [34]: Alcantarilla, P.F.; Bartoli, A.; Davison, A.J. KAZE Features. In Proceedings of the Computer Vision—ECCV 2012, Florence, Italy, 7–13 October 2012.
- [35]: Ming-Hsuan Yang, David J. Kriegman et Narendra Ahuja, “Detecting Faces in Images: A Survey”, IEEE Transactions On Pattern Analysis And Machine Intelligence Vol 24, No, pp 33-58, January 2002.

- [36]: G Yang et T.S.Huang , “Human Faces Detecting in complex background”, Pattern Recognition, Vol 27, No 1, pp 53-63, 1994
- [37]: Grewenig, S.; Weickert, J.; Bruhn, A. From Box Filtering to Fast Explicit Diffusion. In Pattern Recognition, Proceedings of the 32nd DAGM Symposium, Darmstadt, Germany, 22–24 September 2010.
- [38]: Grewenig, S.; Weickert, J.; Schroers, C.; Bruhn, A. Cyclic Schemes for PDE-Based Image Analysis; Technical Report 327; Department of Mathematics, Saarland University: Saarbrücken, Germany, 2013.
- [39]: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2006.
- [40]: Rapid Object Detection using a Boosted Cascade of Simple Features, Conference on Computer Vision and Pattern Recognition. 2001.
- [41]: J. Heinly, E. Dunn, and J. M. Frahm. Comparative evaluation of binary features. In Eur. Conf. on Computer Vision (ECCV).

RÉSUMÉ

L'objectif du manuscrit est de présenter l'application d'une nouvelle variante de descripteur SIFT dans la problématique de détection du visage. Dans notre travail, nous avons examiné les principaux algorithmes de machine-learning, par la suite nous avons introduit le SIFT, dans le cadre de détection de visage. Deuxièmement, nous avons étendu le descripteur présenté SIFT en une décomposition à l'échelle non linéaire en utilisant un filtre de diffusion anisotrope. En utilisant ce nouveau filtre, nous avons éliminé certains points de contrôle ou points clés utilisés dans SIFT traditionnel. Cela peut être supposé par le fait que certains points clés ne sont du fait que de fausses faces. À la fin, nous avons appliqué l'algorithme à un certain nombre de vidéo exemple.

Mots clés : SIFT, SURF, descripteur, KAZE, AKAZE, détection, apprentissage automatique.

ABSTRACT

The aim of the manuscript is to present the application of a new variant of SIFT descriptor in face detection application. A first of all, we have review the main algorithms of the machine learning and then after we have introduced the SIFT, and it back ground into face detection framework. Secondly, we have extended our SIFT descriptor into a non-linear scale decomposition using anisotropic diffusion filter. By using this new filter, we have eliminated some control points or key points used in traditional SIFT. This can be assumed by the fact that some keypoints are due to only false face. At the end we have applied the algorithm to some shearing video.

Key words: SIFT, SURF, descriptor, KAZE, AKAZE, detection, machine learning.