الجمهوريـة الجزائـريـة الديمقراطيـة الشعبيـة
République Algérienne Démocratique et Populaire
وزارة التعليـم العالـي والبحـث العلمـي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Nº Réf :……

Centre Universitaire
Abd elhafid Boussouf Mila

Institut des sciences et de la technologie **Département de Mathématiques et Informatiques**

Mémoire préparé En vue de l'obtention du diplôme de Master

En : **Informatique**

Spécialité : Sciences et Technologies de l'information et de la communication (STIC)

Indexation et recherche d'images par le contenu

Préparé par :

Bensadi Fateh

Bouhebel Radouane

Soutenue devant le jury

Encadrer par **Mezzoudj Saliha ……M.A.B**

Président **TALAI Meriem ……...M.A.A**

Examinateur **BOUMASSATA Meriem ……..M.A.B**

Année universitaire : 2015/2016

Remerciement

Nous remercions avant tout, **Dieu** de nous avoir prodiguée la force morale et physique nous a permis d'achever ce travail.

Nous tenons à exprimer ici notre gratitude au Mm **Mezzoudj Saliha** notre Encadreuse pour tout son investissement à notre égard, la liberté qu'elle nous a laissé dans notre travail, sa grande disponibilité, sa rapidité, son aide inconditionnelle et ce quel que soit le domaine (scientifique ou non)…

Merci à toutes les personnes qui nous ont aidé et entouré durant notre thèse de près ou de loin, surtout nos amis (Abd elghani, Houssem et Djfaflia, Yazid, Ammar, Nouar …).

Enfin, nous n'oublions jamais les noms des auteurs à la fin de ce document. Grâce à leurs excellents articles sur l'Internet, nous retirons les connaissances dans le domaine de l'indexation et la recherche d'images par le contenu.

Dédicace

Tout d'abord, je remercie dieu le tout puissant de m'avoir donné la force, le courage et la santé pour entamer et achever ce mémoire.

Je dédie ce mémoire à ceux qui m'ont toujours aidé, cru en moi en m'encourager et en me soutenant jusqu'à la fin, je dois beaucoup, aux être que j'ai de plus chers, au monde : mes parents j'espère qu'ils seront fiers de moi et de mon travail.

Je le dédie également à :

A mes frères : Aziz, Younes, Kamel, Mohammed et Ahmed et tous les membres de ma famille.

A Notre encadreuse qu'elle nous a orienté beaucoup.

A Tous mes amis : Fateh, Ammar, Fares, Yazid, Nouar, Mouloud, Morad, et à toutes ma promotions.

A tout Personne qui me connait.

Bouhebel radouane

Dédicace

Je dédie ce mémoire de fin d'études

A

Mes parents

en témoignage de ma reconnaissance envers le soutien, les sacrifies et tous les efforts qu'ils ont fait pour mon éducation ainsi que ma Formation

A

Mes sœurs Selma et Aya

A

Mes grands parents

A

Notre encadreuse qu'elle nous a orienté beaucoup

A

Tous mes amis qui ont m'oublié depuis un bon moment et tous mes collègues

A

Notre chère frère Khalil qui nous a quitté très tôt

A

Ceux qui ont participé dans la réalisation de ce travaille soit par leur chargement ou leur conseils

Merci à tous

Fateh Bensadi

Table de matières

Chapitre II : Description des méthodes d'indexation utilisées

Chapitre III : Clustering et analyse de données

Chapitre IV : Description générale du système proposé

Chapitre V : Implémentation et évaluation

ملخص

الهدف الرئيسي من هذه المذكرة هو دراسة الأساليب والطرق الموجودة حاليا في مجال معالجة الصور بصفة عامة وبصفة خاصة في مجال الفهرسة والبحث عن الصور حسب المحتوى وإنشاء نموذج لنظام الفهرسة والبحث عن الصور حسب المحتوى اعتمادا على هده المعارف. ونحن نهتم في مثالنا هذا خاصة على الصور الملونة، والصور الرمادية اللون... الخ.

مع العلم انه يوجد مجموعة كبيرة ومتنوعة من تقنيات البحث عن هذه الصور. اخترنا رسوم بيانية للأنماط المحلية الثنائية الملونة والواصفات الثابتة ضد التحولات الشكلية كمصدر للمعلومات الأساسية لإنشاء النموذج وبعض معايير التشابه.

لقد درسنا أولا كيفية استخراج المعلومات من الصور وبعض معايير التشابه على شكل واصفات بيانية وكذلك طرق فهرسة الواصفات بعد ذلك نستخدم هذه المعلومات في تطبيق النموذج.

النموذج يستقبل صورة، ويؤدي عملية البحث، تم يأتي بصور مماثلة للصورة الأولى معتمدا على الخصائص الموجودة بالصورة.

كلمات مفتاحية: الفهرسة، البحث عن الصور، الديسكريبتور، واصفات بيانية للأنماط المحلية، النقاط المهمة.

Summary

The main objective of this thesis is to study the methods already existing at present generally in the field of image processing and particularly in the field of indexing and image search by content.

In this thesis, we try to use existing methods for making an application, which is interested in indexing and content-based image retrieval. We focus our work specifically with color images and grayscale.

We chose the LBP histograms and SIFT descriptors of color images as a basic source of information and correlation as similarity measures between descriptors.

In the same framework, we tested our own method based on the combination of both LBP and SIFT methods.

In this paper, we first investigate how to extract useful information from a picture to create descriptors using the information contained in the various images of the used base. The classification "Clustering" will be also presented in this thesis.

The application designed receives an input query image, and searches based on the descriptors of the methods that will be used in the search and returns as output the images similar to the query image.

Keywords: indexing, image search, descriptors, LBP, SIFT, features detection.

Résumé

L'objectif principal de ce mémoire est d'étudier les méthodes déjà existantes à l'heure actuelle généralement dans le domaine du traitement d'images et particulièrement dans le domaine de l'indexation et de la recherche d'images par le contenu.

Dans ce travail nous essayons d'utiliser les méthodes existantes pour la réalisation d'une application qui s'intéresse à l'indexation et la recherche d'image par le contenu. Nous nous intéressons dans notre travail plus particulièrement aux images couleurs et en niveaux de gris.

Nous avons choisi les histogrammes LBP et les descripteurs SIFT des images en couleurs comme une source d'information de base et la corrélation comme mesures de similarités entre les descripteurs.

Dans le même cadre nous avons testé notre propre méthode qui se base sur la combinaison des deux méthodes LBP et SIFT.

Dans ce mémoire, nous étudions d'abord comment extraire les informations utile d'une image nécessaire pour la création des descripteurs en utilisant les informations que contient les différentes images de les bases utilisées. La classification « Clustering » va être aussi présentée sain de ce mémoire.

L'application conçue reçoit une image requête en entrée, effectue une recherche basant sur les descripteurs et également les méthodes qui vont être utilisées dans la recherche et retourne en sortie les images similaires à l'image requête.

Mots clés : Indexation, recherche d'images, descripteurs, LBP, SIFT, points d'intérêt.

Liste des Tableaux

Liste des figures

Liste des acronymes

CBIR	Content Based Image Retrieval
SRI	Système de recherche d'information
RSV	retrieval status value
RGB	Rouge, Vert, Blue
HSV	hue, saturation, value
GLDM	Gray Level Dependence Matrix
ACP	Analyse en Composantes Principales
KLT	Karhunen-Loeve transform
SVD	Singular Value Decomposition
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
LBP	Local Binary Pattern
DoG	Difference of Gaussian
IDE	Interactive Development Environment
JRE	Java Runtime Environment
Coil	Columbia Object Image Library

Introduction Générale

Introduction générale

L'évolution numérique à la fin du XXe et le début du XXIe siècle été potentiellement considérable, tous ces innovations dans tous les domaines et surtout dans la télécommunication tel que internet et la photographie numérique ont mené à la création et l'utilisation d'immense bases d'images numérique, qui servent dans le domaine professionnelle tel que (le journalisme, la recherche scientifique, …) ou les particulier qui collectent de grandes quantités des donnés photographique numérique.

Avec le gros grandissement de la masse des donnés photographique, la recherche des images spécifiques ou la gestion manuelle de ces bases a devenu plus en plus difficile à faire, là ou un système d'indexation et recherche d'images doit être conçu pour faire cette tache couteuse en effort et en temps.

A la différence des documents textuels, pour lesquels des méthodes d'indexation et de recherche existent depuis les années 1970 [1], les outils d'analyse et d'interprétation de l'image sont souvent de décalage avec le contenu sémantique, souvent riche de celle-ci. Nous admettons communément deux niveaux d'indexation :

- Le premier niveau dit « numérique » fait référence aux caractéristiques primaires ou bas niveaux telles que la couleur, la forme, la texture, etc.
- Le deuxième niveau dit « sémantique » a trait à l'interprétation de l'image.

Ainsi, les objectifs de notre travail sont :

- D'étudier l'état de l'art des méthodes et des systèmes déjà existants.
- De construire un système d'indexation et recherche d'images par le contenu.
- D'essayer de combiner des différentes méthodes pour avoir des bons résultats

Outre cette introduction, le mémoire se compose de cinq chapitres organisés comme suit :

Nous avons consacré le chapitre 01 pour les concepts de base de la recherche et l'indexation des images par le contenu à savoir l'architecture global du système, les types d'indexation et les méthodes de recherche déjà existantes .

Le chapitre 02 est consacré à la présentation et la définition des différents algorithmes et méthodes d'extraction des informations depuis des images, où ces méthodes seront utilisées dans la conception du système d'indexation et recherche d'image par le contenu que nous allons le proposé et qui va être présenté en détails dans les chapitres suivants.

Dans le chapitre 03, nous aborderons les méthodes de classification et analyse de données lors de la phase de recherche pour bien faciliter et réduire le temps d'exécution d'une recherche par le système.

Dans le chapitre 04, nous allons présenter le système proposé, et parlons de son architecture et les étape que le système suivre durant ces différents phases tel que la phase d'indexation, classification et recherche d'une manière bien détaillé.

Le chapitre 05 revient sur la présentation de notre implémentation. Ainsi, ce chapitre présente les différentes évaluations de notre implémentation ainsi qu'une brève présentation du langage de programmation, la plateforme et les différentes bases d'images utilisé.

Chapitre 01 :

Généralités sur la recherche d'image par contenu

I.1Introduction

La recherche et l'indexation d'image par contenu (CBIR) est un champ richement exploité qui remplace efficacement la recherche textuelle des images qui s'appuient essentiellement sur des annotations des experts du domaine, tandis que la CBIR décrit le contenu visuel par des attributs de bas niveau extraits automatiquement de l'image.

Partant de cela, l'utilisateur peut interroger ce type de système par des images exemples et non plus par des mots clés, le système par la suite se charge de lui retourner des réponses pertinentes dans des délais brefs.

En général, la représentation des images par des caractéristiques visuelles est basée sur trois primitives : la couleur, la forme, la texture. En se basant sur ces caractéristiques, de nombreux systèmes de recherche d'images par le contenu ont été proposés. Ces systèmes de recherche d'images par le contenu sont basée sur un ou plusieurs de ces caractéristiques.

Dans ce chapitre, nous présentons quelque notion de base qui concerne ce domaine d'indexation et recherche d'image pas le contenu.

I.2 Indexation et recherche d'image par contenu visuel

Les systèmes de recherche d'images par le contenu permettent la recherche des images d'une base de données en fonction de leurs caractéristiques visuelles. Ces caractéristiques, encore appelées caractéristiques de bas-niveau, sont la couleur, la texture et la forme, ces derniers sont stockées dans un vecteur numérique appelé descripteur visuel. En général dans ces systèmes, la requête est une image de la base de données, et le résultat de la requête correspond à une liste d'images ordonnées en fonction de la similarité entre leurs descripteurs visuels et celui de l'image requête, en utilisant une mesure de distance. Plus la distance entre les descripteurs visuels de deux images tend vers zéro, plus les images sont considérées comme similaires [01].

I.3 Architecture des systèmes d'indexation des bases d'images

Deux aspects indissociables coexistent dans les systèmes de recherche d'images par le contenu sont l'aspect d'indexation et la recherche.

- ***La phase d'indexation (hors-Ligne) :*** Un système d'indexation comprend généralement deux phases de traitement :

- Indexation logique :

Consiste à extraire et à modéliser les caractéristiques de l'image qui sont principalement la forme, la couleur et la texture. Chacune de ces caractéristiques pouvant être considérée pour une image entière ou pour une région de l'image.

- Indexation physique :

Consiste à déterminer une structure efficace d'accès aux données pour trouver rapidement une information. De nombreuses techniques basées sur des arbres (arbre-B, arbre-R, arbre quaternaire,…) ont été proposées [02].

Pour qu'un système de recherche d'images soit performant, il faut que l'indexation logique soit pertinente et que l'indexation physique permette un accès rapide aux documents recherchés.

- **La phase recherche (En-ligne) :**

Dans cette étape, le système analyse une ou plusieurs requêtes émises par l'utilisateur et lui donne le résultat correspond en une liste d'images ordonnées, en fonction de la similarité ou de la corrélation entre leur descripteurs visuels et celui de l'image requête en utilisant une mesure de distance (à titre d'exemple la distance de Hamming).

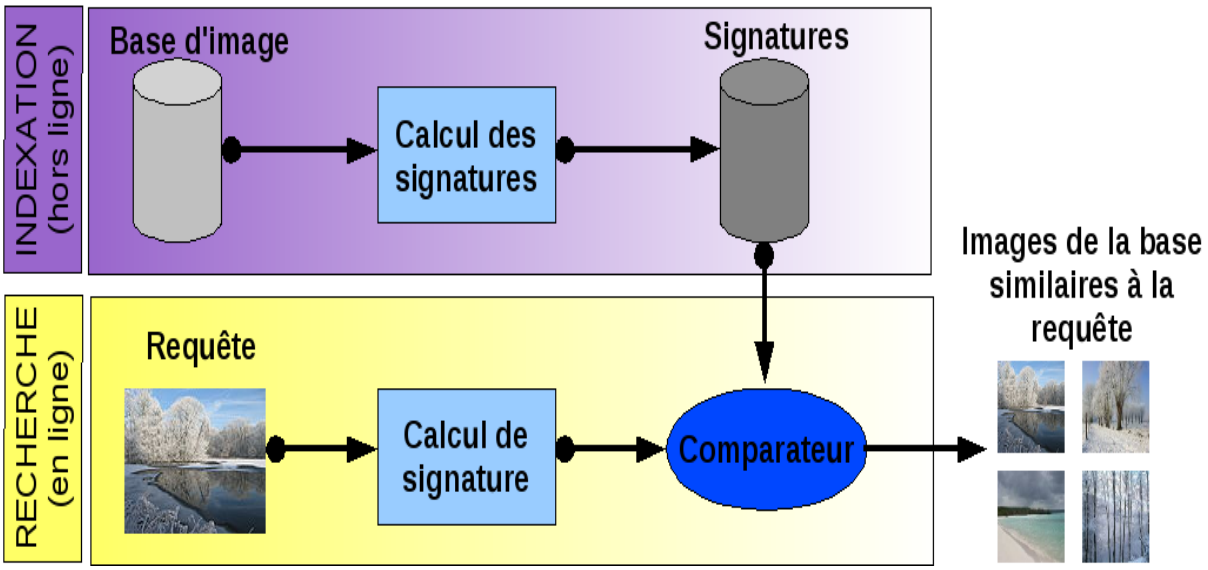


Figure 1.1. L'architecture générale d'un système d'indexation et recherche d'image par le contenu.

I.4 Modèles classiques de recherche d'information pour l'image

Les modèles de représentation d'information (les descripteurs des images) définit dans le cadre de la recherche des images sont ceux tirés de la recherche documentaire.

Globalement, on distingue trois principales catégories de modèles : modèles booléens, modèles vectoriels et modèles probabilistes.

1.4.1 Modèle booléen

Les premiers SRI développés sont basés sur le modèle booléen, même aujourd'hui beaucoup de systèmes commerciaux (moteurs de recherche) utilisent le modèle booléen. Cela est dû à la simplicité et à la rapidité de sa mise en œuvre [03].

Le modèle booléen est basé sur la théorie des ensembles et l'algèbre de Boole. Dans ce modèle, une image est représentée par une liste de descripteurs reliés par l'opérateur. La requête de l'utilisateur est représentée par une expression logique, composée de termes reliés par des opérateurs logiques : ET($\wedge$) , OU ($\vee$) et SAUF ($\neg$).

L'appariement (RSV) entre une requête et une image est un appariement exact, autrement dit si une image implique au sens logique la requête alors l'image est pertinente. Sinon, elle est considéré non pertinente. La correspondance entre image et requête est déterminée comme suit :

$$\mathbf{RSV\ (q,\ d)} = \begin{cases} \mathbf{1} & \textbf{si d appartient à lénsembel décrit par q} \\ \mathbf{0} & \textbf{sinon} \end{cases} \quad \textbf{(I.1)}$$

1.4.2 Modèle vectoriel

La représentation vectorielle des images a été mise en œuvre par GerardSalton pour le système *SMART*, Les images sont représentées par des vecteurs dans un espace vectoriel multidimensionnel, dont les coordonnées sont déterminées par des termes d'index apparaissant dans l'image.

L'hypothèse de base de ce modèle est l'indépendance d'occurrences des termes d'index : la présence d'un terme dans une image ne dépend pas du reste des termes dans cette image.

À chaque image *D* est associé un vecteur $\vec{D} = (dt_1, dt_2, \ldots, dt_n)$, où

$dt_1, dt_2, \ldots, dt_n$ sont les nombres d'occurrences des termes $t_1, t_2, \ldots, t_n$ dans ***D***.

La pertinence de l'image *D* par rapport à une requête *Q*, représentée également par un vecteur dans le même espace, peut être calculée en utilisant une distance vectorielle quelconque : **Sim (D, Q) = ||D, Q||** [04].

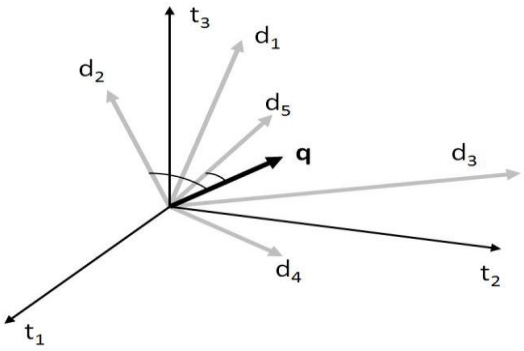


Figure I-2 : Représentation des images dans le modèle vectoriel

L'imaged_5 est plus pertinente que d_1 et d_2 par rapport à la requête q.

I.4.3 Modèle probabiliste

Le modèle probabiliste fait appel à des mesures statistiques, qui cherchent à évaluer la probabilité qu'une image D soit pertinente par rapport à une requête Q, en utilisant des probabilités conditionnelles basées sur les occurrences des termes [04].

Les images retrouvées sont ceux qui ont une forte probabilité d'être pertinents pour la requête et qui ont en même temps une faible probabilité d'être non pertinents. Une première estimation de la distribution des probabilités est améliorée de façon itérative jusqu'à l'obtention d'un ordonnancement final (dans le cas de convergence) des probabilités de pertinence. Ce modèle est coûteux à implémenter et à utiliser à grande échelle. La complexité augmente rapidement avec la taille des collections des images. Tout comme le modèle booléen ou vectoriel, le modèle probabiliste utilise l'hypothèse d'indépendance des termes dans une image**.**

I.5 Extraction des attributs d'images

Des attributs de différents types sont utilisés pour représenter le contenu de l'image. Les attributs sont classés en trois familles principales : la couleur, la texture et la forme.

I.5.1 Les descripteurs de couleurs

La couleur est une caractéristique riche d'information et très utilisée pour la représentation des images. Elle forme une partie significative de la vision humaine. La couleur est devenue la première signature employée pour la recherche d'images par le contenu en raison de son invariance par rapport à l'échelle, la translation et la rotation [05].

Ces valeurs tridimensionnelles font que son potentiel discriminatoire soit supérieur à la valeur en niveaux de gris des images. Une indexation couleur repose sur deux principaux choix : l'espace colorimétrique et le mode de représentation de la couleur dans cet espace[06].

I.5.1.1 Les histogrammes

L'histogramme est défini comme une fonction discrète qui associe à chaque valeur d'intensité le nombre de pixels prenant cette valeur. La détermination de l'histogramme est donc réalisée en comptant le nombre de pixel pour chaque intensité de couleur dans l'image[07].

Cependant, il y a quatre problèmes en utilisent les histogrammes pour l'indexation et la Recherche d'images :

Premièrement : ils sont de grandes tailles, donc par conséquent il est difficile de créer une Indexation rapide et efficace.

Deuxièmement : ils ne possèdent pas d'informations spatiales sur les positions des couleurs. Dans certains cas, il y a des images différentes mais ces images ont les mêmes histogrammes.

Troisièmement : ils sont sensibles à de petits changements de luminosité. C'est-à-dire, c'est difficile pour comparer des images similaires dans des conditions différentes.

Quatrièmement : on ne peut pas faire la comparaison partielle des images (objet particulier dans une image), puisque on doit calculer globalement l'histogramme sur toute l'image.

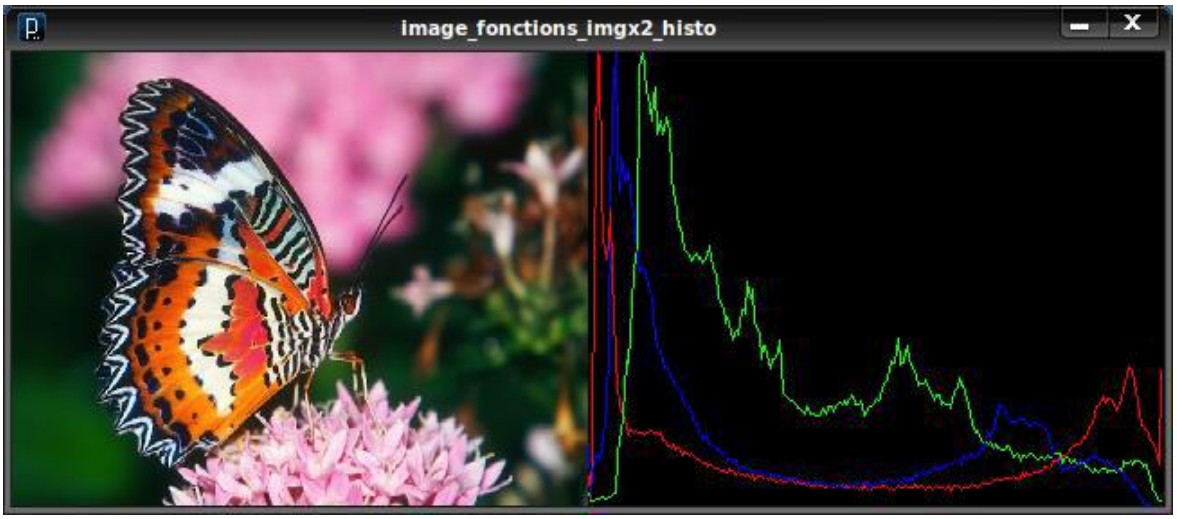


Figure I-3 : Histogramme d'une image.

I.5.1.2 Les espaces de couleurs

Avant de sélectionner un type de description du contenu couleur, il convient de choisir un espace de couleur. Une couleur est généralement représentée par trois composantes. Ces composantes définissent un espace de couleurs. Plusieurs études ont été réalisées sur l'identification d'espaces colorimétriques le plus discriminants mais sans succès car il n'existe pas d'espace de couleurs idéal [8].

Figure 1.4 : Les trois composantes R, V et B de l'image « Lena ».

Il existe plusieurs espaces colorimétriques qui ont chacun certaines caractéristiques intéressantes.

Le modèle RGB propose de coder sur un octet chaque composante de couleur, ce qui correspond à 256 intensités de rouge (28), 256 intensités de vert et 256 intensités de bleu, soient 16777216 possibilités théoriques de couleurs différentes, c'est-à-dire plus que ne peut en discerner l'œil humain (environ 1.6 millions de couleurs). Toutefois, cette valeur n'est que théorique car elle dépend fortement du matériel d'affichage utilisé. La figure 1.4 montre un exemple des trois composantes de l'image « *Lena* ».

Etant donné que le codage RGB repose sur trois composantes proposant la même gamme de valeur, on le représente généralement graphiquement par un cube dont chacun des axes correspond à une couleur primaire comme il est présenté dans la figure 1.5.

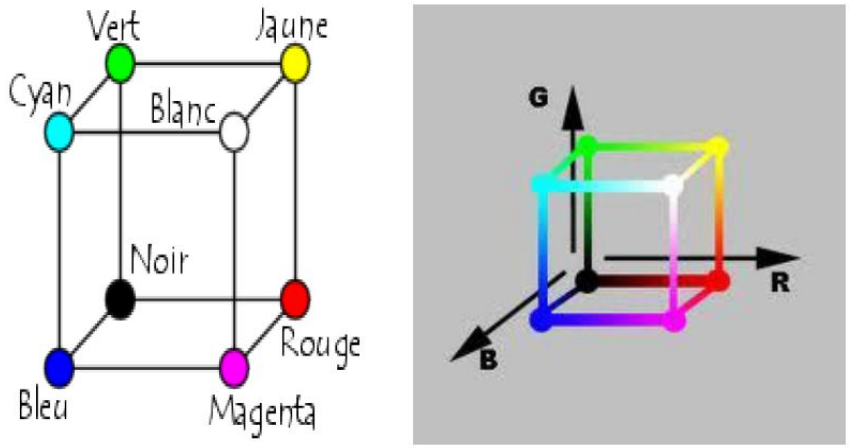


Figure 1.5. Représentation du modèle RGB.

Le modèle HSV consiste à décomposer la couleur selon des critères physiologiques :

- La teinte (en anglais *Hue*) :

La teinte est codée suivant l'angle qui lui correspond sur le *cercle des couleurs* :

1. 0° ou 360° : rouge
2. 2. 60° : jaune.
3. 3. 120° : vert.
4. 4. 180° : cyan.
5. 5. 240° : bleu.
6. 6. 300° : magenta.
7. La teinte varie entre 0 et 360, mais est parfois normalisée en 0–100 %.

-La saturation :

La saturation est l'« intensité » de la couleur :

- elle varie entre 0 et 100 %.
- elle est parfois appelée « pureté ».
- plus la saturation d'une couleur est faible, plus l'image sera « grisée » et plus elle apparaitra fade, il est courant de définir la « désaturation » comme l'inverse de la saturation.

-La luminance :

La valeur est la « brillance » de la couleur :

- elle varie entre 0 et 100 %.
- plus la valeur d'une couleur est faible, plus la couleur est sombre. Une valeur de 0correspond au noir.

La figure 1.6 donne une représentation graphique du modèle HSV, dans lequel la teinte est représentée par un cercle chromatique et la luminance et la saturation par deux axes.

Le modèle HSV a été mis au point dans le but de permettre un choix interactif rapide d'une couleur, pour autant il n'est pas adapté à une description quantitative d'une couleur.

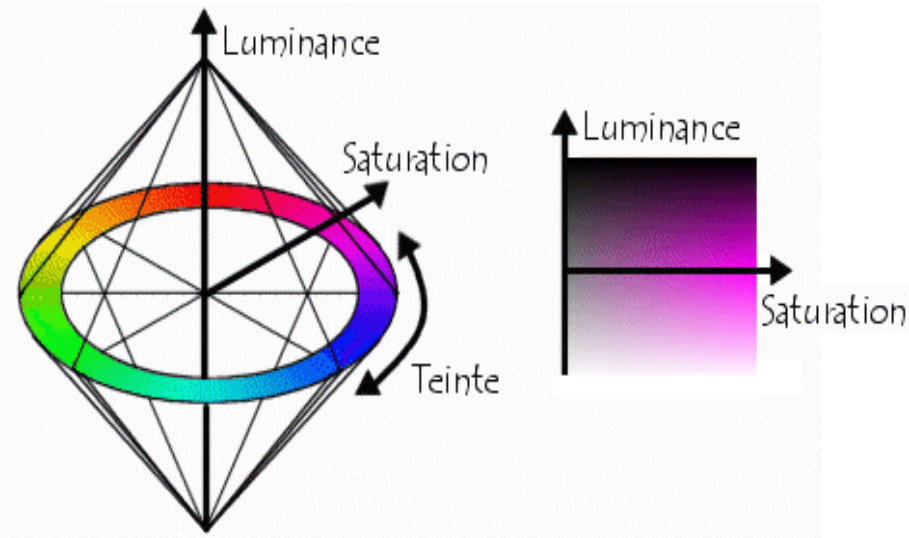


Figure 1.6. Représentation graphique du modèle HSV.

Il existe d'autres modèles naturels de représentation proches du modèle HSV :

- HSB : Hue, Saturation, Bright Ness soit Teinte, Saturation, Brillance en français.
- La brillance : décrit la perception de la lumière émise par une surface.
- HSI : Hue, Saturation, Intensité en français.
- HCI : Hue, Chrominance, Intensité en français.

L'espace CIE XYZ prend en compte la sensibilité de l'œil. La composante Y représente la luminance, X et Z contient l'information de chrominance. Il est rarement utilisé en recherche d'images car il n'est pas uniforme de point de vue humaine. De plus, il n'est pas facile d'interpréter les valeurs de tri stimulus X, Y et Z et d'interpréter les couleurs qu'il représente [09].

La distance de couleur associée à cet espace est la distance euclidienne

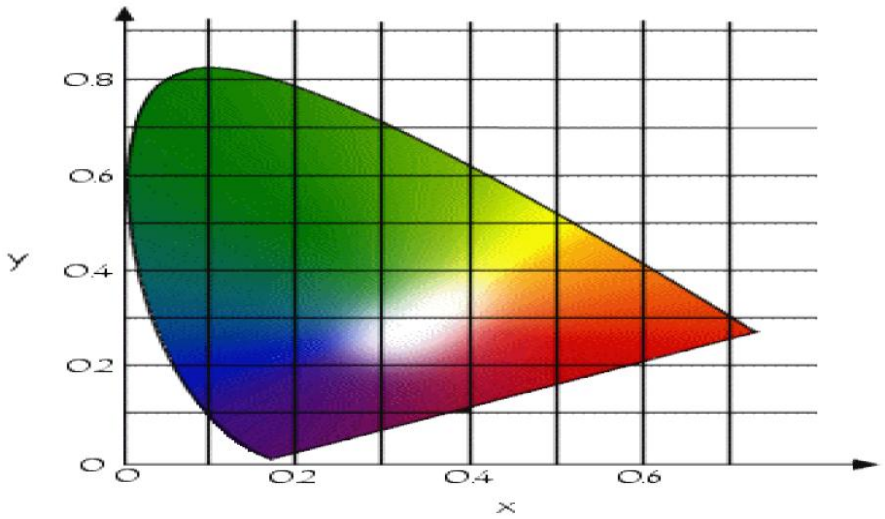


Figure 1.7. Représentation de l'espace couleur XYZ

I.5.2 Les descripteurs de textures

La texture est une information de plus en plus utilisée dans l'indexation et la recherche d'images par le contenu, car elle permet de pallier certains problèmes posés par l'indexation par la couleur, notamment lorsque les distributions de couleur sont très proches. Cependant, il n'existe pas de définition formelle de ce qu'est une texture, et il n'en existe pas non plus de représentation officielle. La figure 1.7 illustre quelques exemples des textures différentes.

Figure 1.7. Différents modèles de texture.

De nombreuses approches et modèles ont été proposées pour la caractérisation de la texture. Parmi les plus connues, on peut citer : les ondelettes, la matrice de co-occurrence.

I.5.2.1 Les ondelettes

À partir de la transformée en ondelettes on peut extraire des attributs de différents types et à différents niveaux de résolution. L'image d'approximation donne des informations sur les régions qui composent l'image, d'une résolution fine à une résolution grossière. Les images de détails donnent des informations horizontales, verticales et diagonales sur l'image.

L'énergie des coefficients d'ondelettes est directement disponible, on la calcule en prenant la somme des carrés des coefficients d'ondelettes. On a ainsi des mesures d'énergie à différents niveaux de résolution [10].

I.5.2.2 La matrice de co-occurence

La texture d'une image peut être interprétée comme la régularité d'apparition de couples de niveaux de gris selon une distance donnée dans l'image. La matrice de **co-occurrence** contient les fréquences spatiales relatives d'apparition des niveaux de gris selon quatre directions : $\theta = 0$, $\theta = \pi/2$, $\theta = \pi/4$ et $\theta = 3\pi/4$. La matrice de **co-occurence** est une matrice carrée n*n où n est le nombre de niveaux de gris de l'image. On définit la matrice des fréquences relatives F par :

F(d, θ) = (f(i, j|d, θ)) où f(i, j|d, θ) représente le nombre de fois où un couple de points séparés par la distance ddans la direction θ a présenté les niveaux de gris g_i et g_j .

I.5.2.3 La méthode de différence de niveaux de gris

La méthode de différence de niveaux de gris GLDM permet de calculer le nombre d'apparition d'une différence de niveaux de gris donnée. Cela revient à calculer des paramètres sur une image de différence entre une image initiale et une image translatée de d.

La GLDM donne un aspect de la texture au sens de la différence de niveaux de gris entre les pixels. Cette différence des niveaux de gris est définie pour chaque pixel d'une région donnée par :

$$|g = |f(x, y) - f(x + dx, y + dy| \quad \textbf{(I.2)}$$

où $f(x, y)$est le niveau de gris au point de coordonnée (x,y), t les coordonnées du vecteur déplacement sont décrites par (dx,dy).

Dans cette technique, on considère que la distribution des valeurs prises par g pour l'ensemble des pixels appartenant à l'objet caractérise la texture. On résume la distribution de g par les paramètres usuels de la statistique. Parmi ces paramètres on peut citer le contraste, la moyenne, l'entropie, le second moment angulaire.

I.5.3 Les descripteurs de la forme

La forme est généralement une description très riche d'un objet. L'extraction d'attribut géométrique a été le fer de lance de la recherche d'image par le contenue dernières années. De nombreuses solutions ont été proposées pour représenter une forme, nous distinguons deux catégories de descripteurs de formes :

- Les descripteurs basés sur les régions.

- Les descripteurs basés sur les frontières.

Les premiers font classiquement référence aux moments invariants et sont utilisés pour caractériser l'intégralité de la forme d'une région. Ces attributs sont robustes aux transformations géométriques comme la translation, la rotation et le changement d'échelle. La seconde approche fait classiquement référence aux descripteurs de Fourier et porte sur une caractérisation des contours de la forme. Nous présentons dans ce qui suit quelques méthodes de description de la forme

I.5.3.1 Les moments géométriques

Les moments géométriques permettent de décrire une forme à l'aide de propriétés statistiques. Ils sont simples à manipuler mais leur temps de calcul est très long [11].

Formule générale des moments :

$$\boldsymbol{m_{p.q} = \sum_{P=0}^{m} \sum_{q=0}^{n} x^p x^q f(x, y)} \textbf{(I.3)}$$

p+ q : l'ordre du moment.

$\boldsymbol{m_{0,0}}$: représente le moment d'ordre 0.

Les deux moments d'ordre 1 $\boldsymbol{m_{0,1}}$et$\boldsymbol{m_{1,0}}$, associés au moment d'ordre 0.

$\boldsymbol{m_{0,0}}$ permettent de calculer le centre de gravité de l'objet.

Les coordonnées de ce centre sont :

$$\mathbf{x_c} = \frac{\mathbf{m_{1,0}}}{\mathbf{m_{0,0}}}$$

$$\mathbf{y_c} = \frac{\mathbf{m_{0,1}}}{\mathbf{m_{0,0}}}$$

Il est possible de calculer à partir de ces moments l'ellipse équivalente à l'objet. Afin de calculer les axes de l'ellipse, il faut ramener les moments d'ordre 2 au centre de gravité :

$$\boldsymbol{m^g_{2,0} = m_{2,0} - m_{0,0}\, x_c^2} \textbf{(I.4)}$$

$$\boldsymbol{m^g_{1,1} = m_{1,1} - m_{0,0}\, x_c y_c} \textbf{(I.5)}$$

$$\boldsymbol{m^g_{2,0} = m_{0,2} - m_{0,0}\, y_c^2} \textbf{(I.6)}$$

Puis on détermine l'angle d'inclinaison de l'ellipse.

$$\boldsymbol{\alpha = \frac{1}{2} arctan \frac{2m^g_{1,1}}{m^g_{2,0} - m^g_{0,2}}} \textbf{(I.7)}$$

I.5.3.2 : Descripteurs de Fourier

Les Descripteurs de Fourier DFs font partie des descripteurs les plus populaires pour les applications de reconnaissance de formes et de recherche d'images. Ils ont souvent été utilisés par leur simplicité et leurs bonnes performances en termes de reconnaissance et facilitent l'étape d'appariement. De plus, ils permettent de décrire la forme de l'objet à différents niveaux de détails. Les descripteurs de Fourier sont calculés à partir du contour des objets. Leur principe est de représenter le contour de l'objet par un signal 1D, puis de le décomposer en séries de Fourier. Les DFs sont généralement connus comme une famille de descripteurs car ils dépendent de la façon dont sont représentés les objets sous forme de signaux.

1.5.4 :Segmentation et points d'intérêt

I.5.4.1 La segmentation

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux des critères prédéfinis. Les pixels sont ainsi regroupés en

régions, qui constituent une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond [12].

La segmentation est une étape primordiale en traitement d'image. À ce jour, il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en trois classes principales:

1. Segmentation fondée sur les contours (edge-based segmentation).
2. Segmentation fondée sur les régions (region-based segmentation).
3. Segmentation fondée sur classification ou le seuillage des pixels en fonction de leur intensité.

I.5.4.2 Point d'intérêt

Les points d'intérêt sont des régions de l'image riches en termes de contenu de l'information locale et stables sous des transformations affines et des variations d'illumination. Ils sont des indicateurs des régions susceptibles de contenir un objet, et en même temps des parties importantes de l'objet.

Trois types d'approche pour l'extraction de points d'intérêt :

- **Approches contours :** les contours d'une image sont d'abords détectés, puis les points d'intérêt sont extraits le long des contours en considérant les points de courbures maximales ainsi que les intersections de contours.
- **Approches intensité :** la fonction d'intensité est utilisée pour extraire directement des images les points de discontinuité.
- **Approche à base de modèle :** les points d'intérêt sont identifiés dans l'image par mise en correspondance de la fonction d'intensité avec un modèle théorique.

I.6 *Indexation multidimensionnelle*

Dans le cadre de bases d'images volumineuses, l'organisation des descripteurs de contenu pose deux problèmes lors de la phase d'indexation :

- ***Haute dimensionnalité*** *:* Le nombre de descripteurs, notamment la taille du descripteur, qui doit être le plus minimale possible.
- ***Mesure de similarité non-euclidienne :*** La technique d'indexation doit être capable de supporter d'autres mesures de similarité du fait que la distance euclidienne ne peut discriminé efficacement certain contenus visuels.

Une solution proposée pour pallier à ces deux problèmes est d'effectuer une réduction de la dimension du descripteur, ensuite adopter une technique d'indexation multidimensionnelle

Supportant d'autres mesures de similarité plus efficaces que la distance euclidienne. Position dans un espace multidimensionnel.

Les avantages d'un index spatial sont l'efficacité en temps et en espace pour la recherche d'objets et la mise à jour dynamique et efficace. Les données peuvent être placées sans organisation particulière autre que leur stockage par page de mémoire**.**

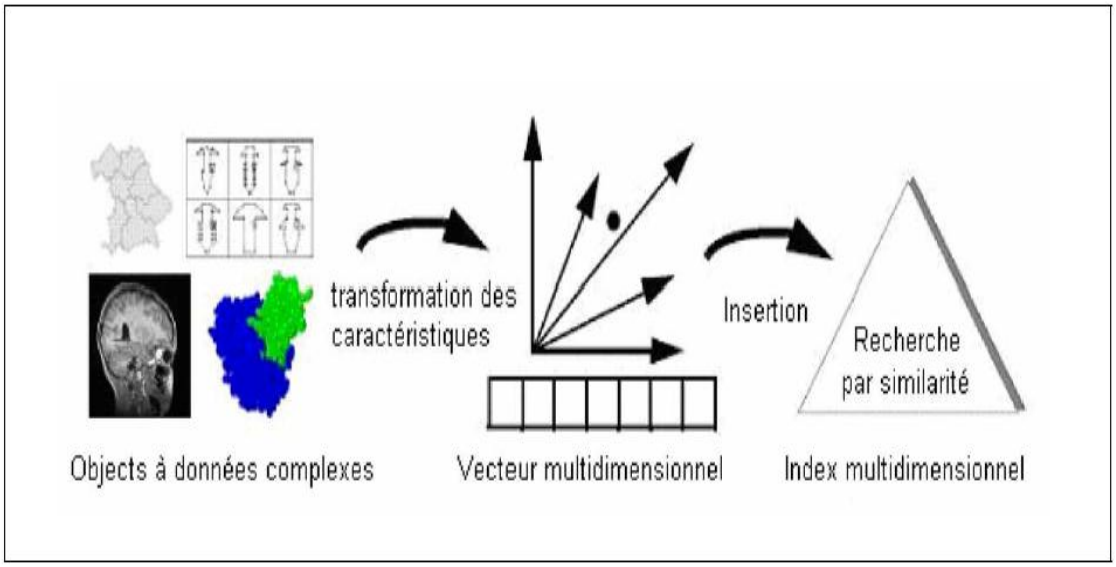


Figure 1.8 : Principe général de l'indexation multidimensionnel

1.6.1 Réduction de la dimension

Avant d'utiliser les techniques d'indexation, il est recommandé d'effectuer une réduction d'espace en premier lieu. Deux approches proposés dans la littérature : Karhunen-Loeve transform (KLT) [13], et le column-wise Clustering.

KLT et ses variétés dans la reconnaissance faciale, eigenimage, ses variétés dans l'analyse de données, Analyse en Composantes Principales (ACP) qui s'applique sur l'ensemble des descripteurs de la base d'images afin de construire une nouvelle base vectorielle de taille réduite, dans laquelle sont ensuite projetés les descripteurs originaux.

Nombreux travaux ont été élaborés dans le champ de la réduction d'espace ont conduit à des améliorations de la KLT, dans [14] les auteurs ont mis au point une décomposition faible rang de valeur singulière : (SVD) algorithme de mise à jour efficace et numériquement stable dans l'utilisation de KLT.

De plus que la KLT, le regroupement des colonnes (column-wise) ou des lignes (row-wise) est une autre technique utilisé dans la reconnaissance de forme, l'extraction des informations.

Elle consiste à regrouper des objets similaires (motifs, signaux, documents) ensembles pour effectuer la reconnaissance ou le groupement. Cette approche est expérimentalement désignée comme simple et efficace.

Cependant l'inconvénient de cette réduction est la perte de toute information sémantique des descripteurs.

I.6.2 Techniques d'indexation multidimensionnelle

I.6.2.1 Techniques basées sur le partitionnement des données

Ces techniques dérivent toutes du R-Tree. Elles procèdent par regroupement des vecteurs selon leur proximité dans l'espace. Chaque groupe de vecteurs est englobé dans une forme géométrique simple à manipuler (hyper-sphère,hyper-rectangle,etc.). Le tout est organisé sous forme d'un arbre dans lequel les vecteurs sont stockés dans les feuilles alors que les formes englobâtes sont stockées dans les nœuds internes. On trouve dans cette famille des méthodes le R*-Tree, le X-Tree [15].

I.6.2.2 Techniques basées sur le partitionnement de l'espace

Les techniques basées sur le partitionnement de l'espace sont plus simples à gérer que celles basées sur le partitionnement des données, et de plus aucun chevauchement n'existent entre les cellules. Le principe est de partitionner l'espace sans prendre en compte les données. Les cellules peuvent être gérées de deux manières différentes :

- soit elles sont organisées dans un arbre (K-D-B-TREE, LSD-TREE...),
- soit elles sont adressées par une fonction de hachage (GRIDFILE).

I.7 *Type de requête*

Il existe trois façons de faire une requête dans un système d'indexation et recherche des images : soit une requête par mots clés, soit une requête par esquisse, soit une requête par exemple.

1.7.1 Requête par mot clé

Les images sont recherchées suivant un ou plusieurs critères, par exemple trouvé Les images contenant 80% de rouge. Donc, le système se base sur l'annotation manuelle et textuelle d'images.

1.7.2 Requête par esquisse (dessin)

Dans ce cas, le système fournit à l'utilisateur des outils permettant de construire une esquisse (dessin) correspondant à ses besoins. L'esquisse fournie sera utilisée comme exemple pour la recherche. L'esquisse peut être une ébauche de forme ou contour d'une image entière ou une ébauche des couleurs ou textures des régions d'une image.

L'utilisateur choisira, en fonction de la base d'images utilisée, de ses besoins et préférences, l'une ou l'autre de ces représentations. Cette technique présente l'inconvénient majeure qu'il est parfois difficile pour l'utilisateur de fournir une esquisse, malgré les outils qui lui sont fournis.

1.7.3 Requête par l'exemple

Pour les systèmes de recherche d'image à base d'exemple, l'utilisateur, pour Représenter ses besoins, utilise une image (ou une partie d'image) qu'il considère similaire aux images qu'il recherche. Cette image est appelée image exemple ou requête.

L'image exemple peut soit être fournie par l'utilisateur, soit être choisie par ce dernier dans la base d'images utilisée. Cette technique est simple et ne nécessite pas de connaissances approfondies pour manipuler le système.

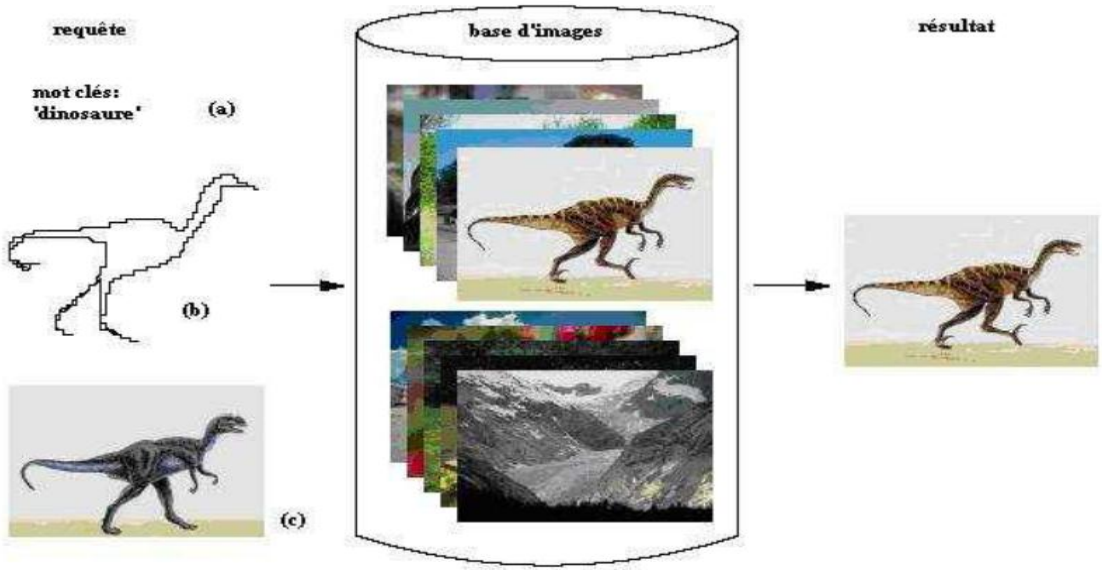


Figure 1.9 : représentation de différents types de requête utilisée pour recherche d'image

I.8 *Mesures de similarité*

Les mesures de similarité ne permettent pas de juger si une image ressemble à une autre, mais elles permettent de mesurer une distance mathématique et de déterminer si elles sont équivalentes où indépendantes. Donc dans notre cas, ces mesures comparent les distances des caractéristiques de l'image de la requête avec celles de la base de données[16].

Il existe plusieurs de ces mesures qui font cette comparaison, mais les plus utilisées dans la recherche d'images sont :

I8.1 Distances géométriques

Le premier type de mesures de distances de similarité correspond aux distances géométriques entre vecteurs. Dans ce cas, on parle de distances car ces mesures ont la propriété de respecter les axiomes des espaces métriques.

Définition des espaces métriques

Maurice Fréchet (1878-1973) définit un espace métrique E comme un ensemble non vide doté d'une application d, appelée distance, de Ex E dans R^+vérifiant les axiomes suivants :

∀x, y, z ∈ E

1- $\mathbf{d(x,y) = 0 \Leftrightarrow x = y}$ **(identité)**

2- $\mathbf{d(x,y) = d(y,x)}$ **(Symétrie)**

3- $\mathbf{d(x,y) + d(y,z) \geq d(x,z)}$ **(Inégalité triangulaire)**

L'axiome de l'inégalité triangulaire est calqué sur la structure métrique du plan muni de la distance euclidienne usuelle, définie dans un repère orthogonal utilisant la même unité sur chaque axe par :

$$\boldsymbol{d(A,B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}} \textbf{(I.8)}$$

Les métriques de Minkowski (ou normes $\mathbf{L_p}$) sont les distances géométriques les plus courantes.

Leur forme générale est la suivante :

$$\boldsymbol{d_{Mink}(I_1, I_2) = \left[\sum_{i=1}^{N} (I_1(i) - (I_2(i))^p\right]^{\frac{1}{p}}} \textbf{(I.9)}$$

I.8.2 Distances entre distributions

L'image peut être considérée comme une variable aléatoire dont les vecteurs d'attributs des pixels sont les réalisations. Le problème de mesure de similarité se ramène alors à déterminer si les réalisations correspondant aux deux images sont issues de la R-néme distribution de probabilités. Nous parlons alors d'approche statistique de la mesure de similarité.

Issue de la théorie de l'information, la divergence de Kullback-Leibler permet de mesurer la distance de similarité (entropie mutuelle) de deux distributions de probabilités :

$$\boldsymbol{d_{kull} = \mathrm{k}(I_1, I_2) = \sum_{i=1}^{n} I_1(i) \log \frac{I_1(i)}{I_2(i)}} \textbf{(I.10)}$$

Cette mesure de distances de similarité est applicable pour la recherche d'images, toutefois, elle ne vérifie pas l'axiome de symétrie et n'est donc pas une distance. Comme suggéré par Puzicha et al. Il est plus intéressant de considérer la distance de Jeffrey vérifiant les axiomes des espaces métriques. Cette métrique est numériquement stable et fait preuve de plus de robustesse au bruit :

$$\boldsymbol{d_{Jeff} = \sum_{i=1}^{n} I_1(i) \log \frac{I_1(i)}{m(i)} + I_2(i) \log \frac{I_2(i)}{m(i)}} \textbf{(I.11)}$$

Avec $\boldsymbol{m(i) = \frac{I_1(i) + I_2(i)}{2}}$

Les tests d'hypothèses sont également adaptés pour le calcul de similarité entre images dans le cadre statistique. Le test du X2, appliqué sous hypothèse gaussienne des distributions, est une alternative présentée comme performante dans la littérature :

$$\boldsymbol{dx^2(I_1, I_2) = \sum_{i=1}^{n} \frac{(I_1(i) - I_2(i))^2}{(I_1(i) + I_2(i))^2}} \textbf{(I.12)}$$

I.8.3 L'intersection d'histogramme

La méthode d'intersection d'histogrammes a été proposée par Swain et Ballard. Elle consiste en la mesure de la partie commune entre deux histogrammes H1 ETH2. Elle est calculée par l'expression suivante :

$$\boldsymbol{D(h, g) = \frac{\sum_{i=1}^{n} \min(H1(i), H2(i))}{\sum_{i=1}^{n} H2(i)}} \textbf{(I.13)}$$

où n est le nombre de valeurs de chaque histogramme [17].

I.9. Critère de performance d'une méthode de recherche d'information

On se trouve devant trois critères principaux pour évaluer le la performance d'une méthode de recherche d'information qui sont :

1.9.1 Le temps :

Le temps de réponse d'une méthode de recherche d'information doit être le plus minimum possible, de plus le temps de réponse est petit le plus la performance est bonne.

1.9.2 La qualité des résultats :

La qualité des résultats rendus par la méthode de recherche est la plus critère focalisé durant la phase de conception et d'évaluation de la méthode, car une méthode de recherche doit toujours retourne des résultats correctes.

1.9.3 La masse des ressources utilisées :

Une méthode de recherche doit pas être couteuse en ressource utilisée durant son exécution, prenons un exemple de mémoire alloué.

I.10 Conclusion

Dans ce chapitre nous avons essayé de décrire les concepts de base pour la construction d'un système d'indexation et de recherche d'images par le contenu. Nous avons également présenté brièvement les différents modèles de recherche d'information pour l'image, ainsi que l'extraction des attributs d'images comme l'attribut couleur.

Les systèmes de recherche d'images par le contenu permettent de rechercher les images d'une base de données en fonction de leurs caractéristiques visuelles. Dans ces systèmes, la requête est une image et le résultat de la requête correspond à une liste d'images ordonnées en fonction de la similarité estimée généralement par le calcul de la distance (à l'aide de l'intersection des histogrammes par exemple) entre le descripteur de l'image requête et ceux des images indexées dans la base.

Chapitre 2 :

Description des méthodes d'indexation utilisées

I.1 Introduction

Après avoir présenté les concepts de base utilisés dans la conception d'un système d'indexation et recherche d'image par le contenu, dans ce chapitre on va passer vers une présentation de quelques méthodes utilisées dans la détection des points d'intérêt et l'extraction des descripteurs qui les correspondent.

D'un point de vue générale, on va parler de trois types de descripteurs principaux qui ont été beaucoup plus utilisé récemment dans la communauté d'indexation et recherche d'images par le contenu, tel que : la méthode SIFT [18], et SURF [20] et LBP [21].

Ce chapitre sera divisé en trois parties, la première présente une description de SIFT, la seconde, une description de SURF et la dernière une description de LBP.

II.2 Description de la méthode SIFT

La méthode des SIFT (en. **Scale-Invariant Feature Transform**, fr.**transformation de caractéristiques visuelles invariante à l'échelle**), est une méthode développée par David Lowe en 2004 [18], permettant de transformer une image en ensemble de vecteurs de caractéristiques qui sont invariants aux transformations géométriques usuelles (homothétie, rotation).

Elle combine les DoG qui sont invariants en translation, rotation et mise à l'échelle avec un descripteur basé sur les distributions d'orientations de gradient qui est plus robuste aux changements d'illumination et de points de prise de vue.

Les étapes de calcul utilisées pour générer l'ensemble des descripteurs de l'image sont décrites ci-dessus :

1. Détection d'espace d'échelle extrême :

La première étape c'est la recherche sur toutes les échelles et les emplacements de l'image, elle est appliquée de manière efficace en utilisant une fonction de DOG (Difference-of-Gaussian) pour identifier les points d'intérêt potentiels qui sont invariantes à l'échelle et l'orientation.

2. Localisation des points d'intérêts :

A chaque emplacement candidat, un modèle détaillé est ajusté pour déterminer l'emplacement et l'échelle. Les points d'intérêts sont choisis en mesure de leur stabilité.

3. Affectation d'orientation :

Une ou plusieurs orientations sont affectées à chaque emplacement d'un point d'intérêt basé sur les directions de gradient d'image locale. Toutes les opérations à venir sont effectuées sur les données d'image transformées par rapport à l'orientation attribuée, à l'échelle, et un emplacement pour chaque caractéristique, garantissant ainsi l'invariance des points détectés à ces transformations.

4. Calcul du descripteur de point d'intérêt :

Les gradients locaux d'image sont évalués à l'échelle détectée dans la région autour de chaque point d'intérêt, puis transformés en une représentation qui permet des niveaux significatifs de la déformation locale et le changement d'illumination.

II.1.1 Détection d'espace d'échelle extrême

L'espace échelle de l'image est une représentation de celle-ci à différentes résolutions et échelles.

Lowe propose d'utiliser la fonction Différence de Gaussiennes (DoG en anglais) pour détecter les extrema dans l'espace échelle. Cette fonction est définie comme la différence de deux images filtrées par un noyau gaussien, séparées par un facteur k :

$$\boldsymbol{D(x; y; \sigma) = L(x; y; k\sigma) - L(x; y; \sigma)} \textbf{(II.1)}$$

Les images L(x; y; σ) sont le résultat d'une convolution de l'image I(x; y) traitée par un noyau gaussien :

$$\boldsymbol{L(x; y; \sigma) = G(x; y; \sigma) * I(x; y) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2+y^2}{2\sigma^2}} * I(x; y)} \textbf{(II.2)}$$

Pour construire D(x ;y ;σ), Lowe propose de calculer avant la pyramide de gaussiennes, par convolution successive de l'image par des noyaux gaussiens pour produire des images séparées par une constante k dans l'échelle espace.

Il divise chaque octave (i.e : on doubleσ) par un nombre entier s d'intervalles, tel que k = $2^{1/s}$. Cela veut dire qu'on doit calculer s + 3 images filtrées pour chaque octave pour avoir les DoG nécessaires à chaque octave. La différence entre images filtrées successives est calculée pour construire une pyramide de DoG qui représente l'espace d'échelle sur lequel on va extraire les points.

Après le calcul de chaque octave, on sous-échantillonne l'image filtrée à l'échelle 2σ pour commencer l'octave suivante.

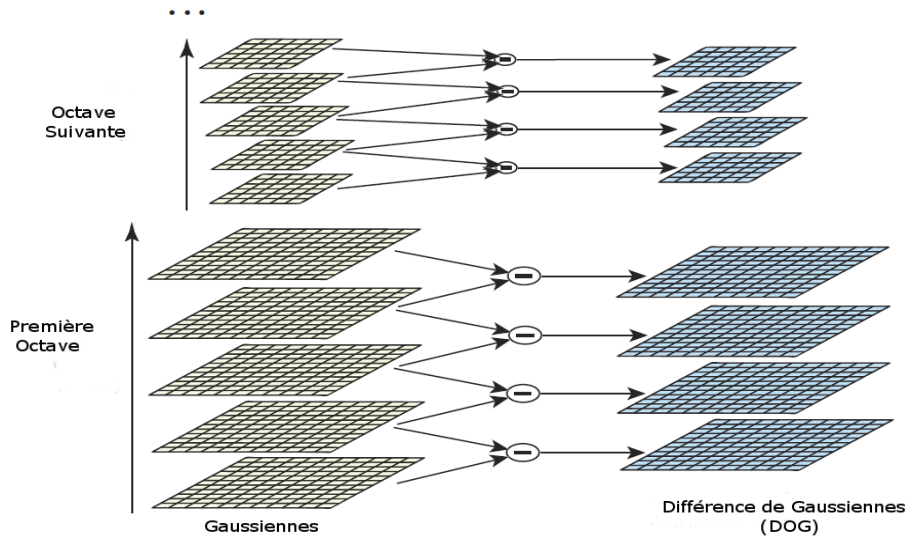


Figure 2.1 : Construction de la fonction Différence de Gaussiennes (DoG).

Pour chaque octave de l'espace échelle, l'image initiale est successivement convolutée par une gaussienne pour produire les différentes images à échelle. Ces images sont montrées à gauche. La différence entre images à échelle successive est calculée pour produire la DoG correspondante comme montré à droite.

Après chaque octave, l'image gaussienne est sous-échantillonnée d'un facteur 2, et le procès recommence.

Les points d'intérêt retenus correspondent aux extrema locaux de D(x; y; σ). Chaque point est comparé à ses 8 voisins dans son propre niveau et aux 9 voisins des échelles supérieure et inférieure .On à besoin d'un total de 26 comparaisons. Si la valeur du

pixel est supérieure ou inférieure aux tous valeurs des pixels testés, on retient le point si non le point va pas être considérer comme candidat d'être un point d'intérêt.

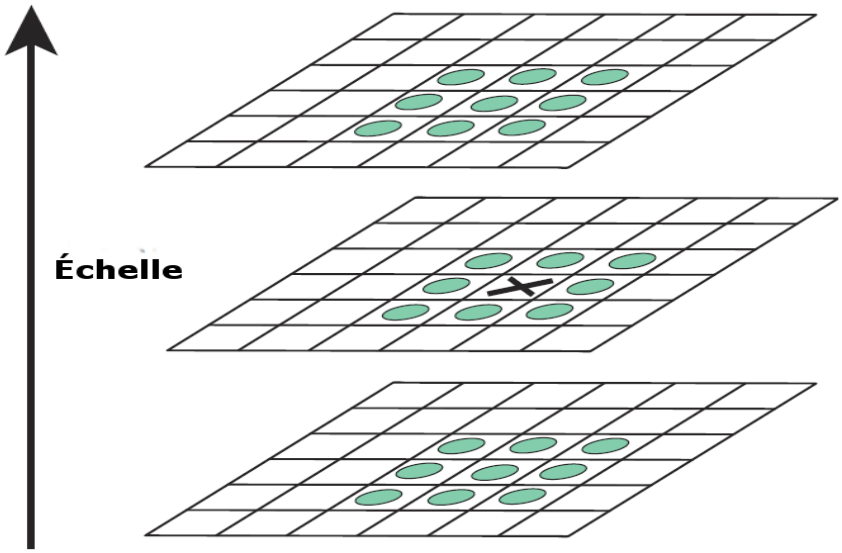


Figure 2.2: Détection des extrema. Les maxima et minima des images des différences de Gaussienne sont détectés on comparant un pixel (marqué par X) à ses 26 voisins dans des régions de 3x3 dans l'échelle courante et adjacentes (marquée par des cercles).

II.1.2 Localisation des points d'intérêt

La deuxième partie de la procédure d'extraction de caractéristiques consiste à localiser de façon précise les points d'intérêts. Pour cela, Lowe propose d'affiner les données acquises en décrivant ≪ l'environnement ≫ de chacun des points d'intérêts. Cette opération permet de rejeter des points instables.

En effet, il est possible d'effectuer une interpolation des coordonnées des points où se trouvent les extremums. Brown et Lowe (2002) utilise un développement de Taylor a l'ordre 2 au point candidat, de la fonction D(x) (différence de gaussiennes) avec x= (x, y, σ) T ou x est un point candidat sélectionné dans l'étape précédente.

$$\boldsymbol{D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x} \textbf{(II.3)}$$

L'extremum localisé $\hat{x}$ est déterminé en dérivant l'expression et en égalisant à zéro, d'où :

$$\boldsymbol{\hat{x} = \frac{-\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}} \textbf{(II.4)}$$

Si l'offset $\hat{x}$ est supérieur à 0,5 dans une des directions on recommence l'interpolation autour du point plus proche. L'offset final $\hat{x}$ est ajouté aux coordonnées des points interpolés pour déterminer la position de l'extrema. La valeur de la fonction à l'extrema D($\hat{x}$) est utile pour rejeter des extrema instables par faible contraste. En cette position on a :

$$\mathbf{D}(\hat{\mathbf{x}}) = \mathbf{D} + \frac{\mathbf{1}}{\mathbf{2}}\frac{\partial \mathbf{D}^{\mathbf{T}}}{\partial \mathbf{x}}\hat{\mathbf{x}} \quad \textbf{(II.5)}$$

La valeur de $|D(\hat{x})|$sera petite pour des points où le contraste est faible. Un filtrage par seuillage permet de rejeter ce type de points.

Finalement les points détectés sur des contours peuvent être rejetés en analysant sa courbure à partir de la matrice Hessienne définie comme :

$$\boldsymbol{H} = \begin{Bmatrix} \boldsymbol{D_{xx}} \boldsymbol{D_{xy}} \\ \boldsymbol{D_{xy}} \boldsymbol{D_{yy}} \end{Bmatrix} \quad \textbf{(II.6)}$$

À partir des valeurs propres on peut détecter si un point est un coin ou un contour.

Si une des valeurs propres est très supérieure à l'autre, il s'agit d'un contour (Forte variation uniquement dans un sens),

Sinon il s'agit d'un coin (Forte variation dans tous les sens).

À partir de la relation entre les valeurs propres $\boldsymbol{r} = \frac{\boldsymbol{\lambda_1}}{\boldsymbol{\lambda_2}}$on peut Filtrer les points détectés sur un contour. À partir de la trace et du déterminant de cette matrice on peut écrire :

$$\frac{\mathbf{T_r(M)^2}}{\mathbf{Det(M)}} = \frac{(\boldsymbol{\lambda_1}+\boldsymbol{\lambda_2})^2}{\boldsymbol{\lambda_1}\boldsymbol{\lambda_2}} = \frac{(\mathbf{r}\boldsymbol{\lambda_1}+\boldsymbol{\lambda_2})^2}{\mathbf{r}\boldsymbol{\lambda_2}^2} = \frac{(\mathbf{r}+\mathbf{1})^2}{\mathbf{r}} \quad \textbf{(II.7)}$$

Qui dépend uniquement du ratio de valeurs propres r. Ce ratio augmente quand r augmente, on peut donc Filtrer les points où ce ratio est inférieure à un seuil.

II.1.3 Affectation des orientations

Pour la propriété d'invariance par rotation, Lowe propose de donner à chaque point une orientation principale, ce qui permet de décrire les points relativement à cette orientation.

L'orientation des points est calculée à partir de l'orientation du gradient dans une région autour du point d'intérêt. On utilise l'échelle du point pour choisir l'image L(x ; y ;σ) plus proche, afin de réaliser les calculs dans un contexte indépendant de l'échelle. On calcule alors pour tous les points la norme et l'orientation du gradient en utilisant les différences de pixels :

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

$$\boldsymbol{\sigma}(\mathbf{x};\ \mathbf{y})\ =\ \mathbf{tan}^{-\mathbf{1}}(((\mathbf{L}(\mathbf{x};\ \mathbf{y}\ +\ \mathbf{1}) - \mathbf{L}(\mathbf{x};\ \mathbf{y} - \mathbf{1}))/((\mathbf{L}(\mathbf{x}\ +\ \mathbf{1};\ \mathbf{y}) - \mathbf{L}(\mathbf{x} - \mathbf{1};\ \mathbf{y}))) \quad \textbf{(II.9)}$$

Un histogramme est formé à partir de l'orientation des gradients des points dans une région autour du point d'intérêt. L'histogramme utilise 36 bins couvrant les 360º du spectre d'orientations. Chaque point est pondéré par la norme de son gradient et par une fenêtre gaussienne circulaire centrée au point et d'écart type de 1.5fois l'échelle du point.

L'orientation du point correspond au pic maximal de l'histogramme. Tout pic supérieur à 80% de la valeur de ce pic est aussi retenu et génère un nouveau point à la même échelle et la même position.

Finalement, l'orientation est raffinée en ajustant une parabole aux 3 valeurs de l'histogramme plus proches du pic.

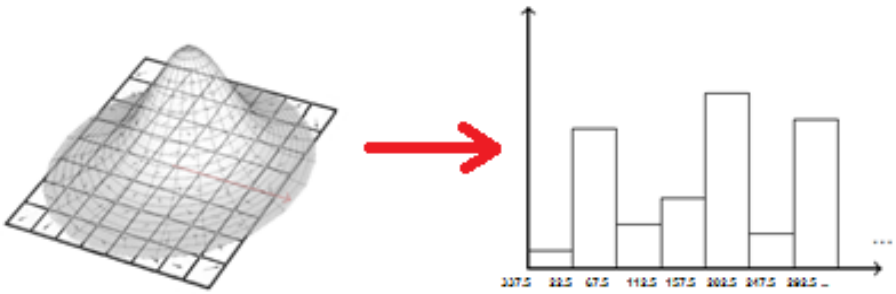


Figure 2.3 : création d'histogramme des orientations à partir de la fenêtre du gradient orienté

II.1.4 Descripteur local de l'image

Un descripteur de point d'intérêt est créé d'abord en calculant la norme du gradient et l'orientation de chaque point dans une région autour du point d'intérêt. Concrètement, Lowe propose de prendre une fenêtre 16x16 autour du point d'intérêt.

Cette fenêtre est divisée en 16 blocs de 4x4. A l'intérieur de chacun de ces blocs 4x4, on calcule l'orientation et la norme des gradients. Ces orientations sont ensuite mise dans un histogramme à 8 graduations (360°/45°) et elles sont pondérées par la norme du gradient au point considéré.

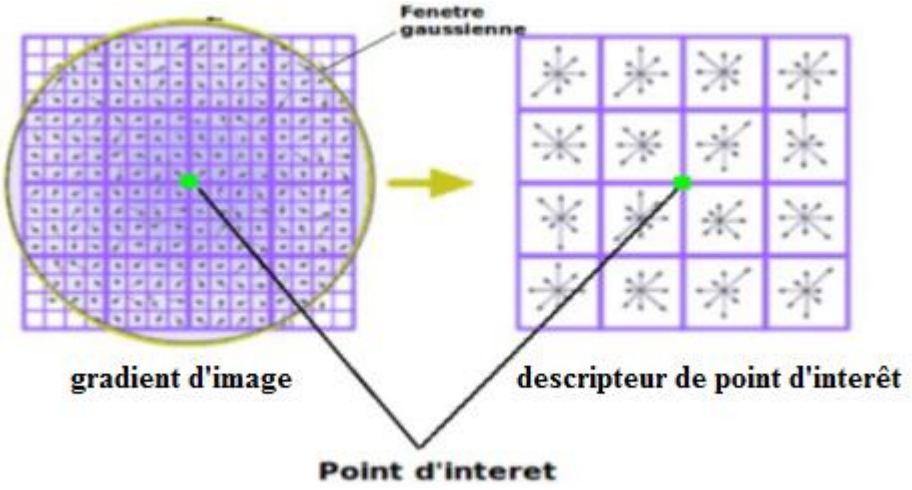


Figure 2.4 : Le descripteur du point d'intérêt est obtenu en calculant la magnitude du gradient et l'orientation de chaque point échantillonnant dans une région autour du point d'intérêt.

L'image des vecteurs gradients ci-dessus (à gauche) de 16x16 pixels, permet d'établir un descripteur4x4. Chaque case du descripteur correspond à l'image du comportement d'un voisinage 4x4 d'un point clé.

Après toutes ces étapes tous nos points d'intérêts se sont vus attribuer un vecteur de caractéristique (un descripteur).

On dispose maintenant, pour caractériser une image, de vecteurs descripteurs à 128 éléments car on a bien 16 histogrammes * 8 directions possibles.

Ces descripteurs sont invariants par mise à l'échelle, rotation et partiellement à l'illumination [19].

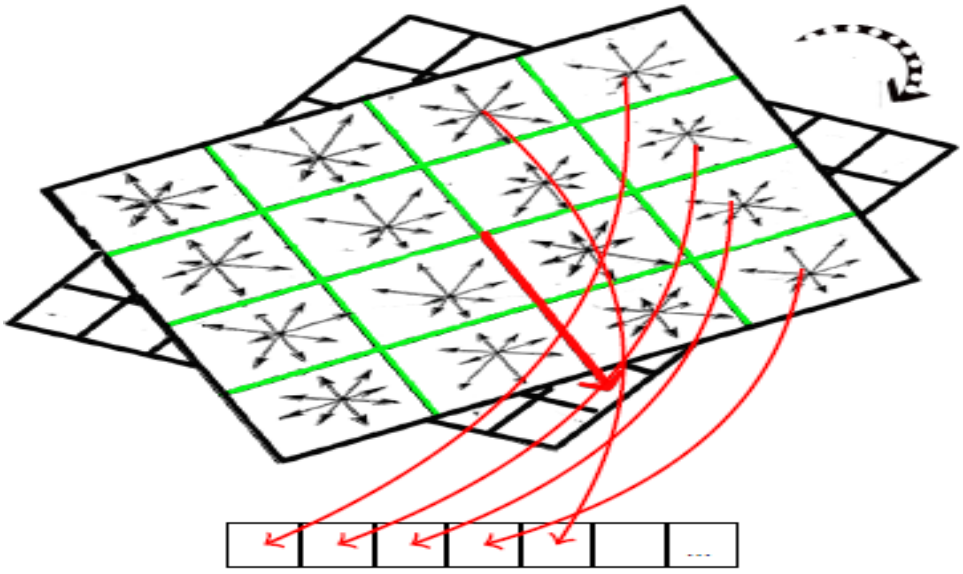


Figure 2.5 : vecteur finale du descripteur

II.3 Description de la méthode SURF

En SURF **Speeded Up RobustFeatures** ,fr. **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 **[20].**

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**.

Les étapes de cette méthode comportent comme les SIFT sont deux étapes :

- Extraction des points d'intérêts
- calcul des descripteurs du point candidat

II.2.1 Détection des points d'intérêt

La méthode des SURF utilise le **FAST-HESSIEN** pour la détection de points d'intérêts et une approximation des **ondelettes de Haar** pour calculer les descripteurs.

Le fast-Hessien se fonde sur l'étude de la matrice HESSIEN :

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

Où $L_{ij}(x,y,\sigma)$est la dérivée seconde suivant les directions en i et en j de L avec :

L(x,y, σ) = G(x,y, σ) ∗ I (x,y)

Où,

$$\boldsymbol{G(x,y,\sigma)} = \frac{\mathbf{1}}{\boldsymbol{\sigma\sqrt{2\pi}}} \boldsymbol{e}^{\frac{\left(\boldsymbol{x^2+y^2}\right)}{\boldsymbol{2\sigma^2}}} \textbf{(II.11)}$$

Et L est l'image de départ.

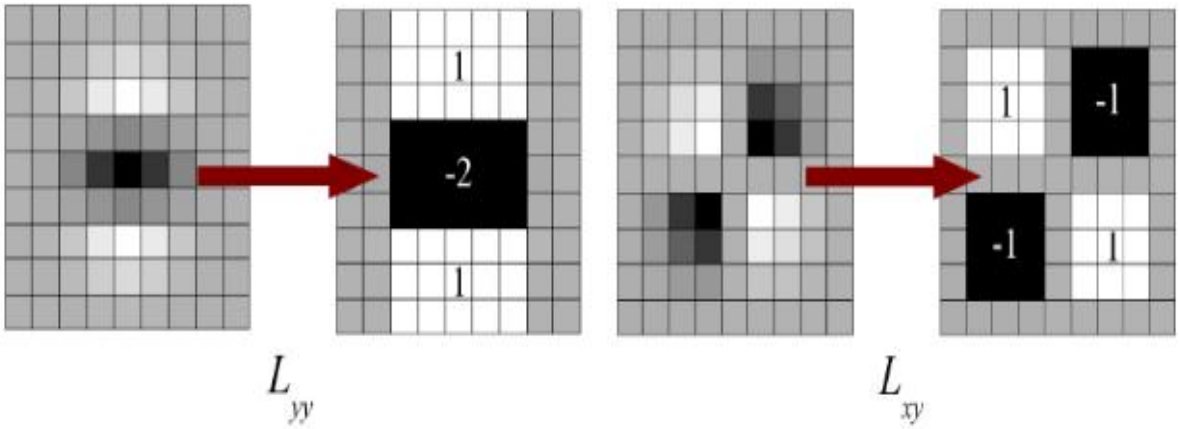


Figure 2.6 : Seconde dérivation avec un filtre moyen

La maximisation du déterminant de cette matrice permet d'obtenir les coordonnées des points d'intérêts à une échelle donnée. Cette étape apporte une invariance des points d'intérêts par rapport à la mise à l'échelle.

Le déterminant est défini ainsi :

$$\mathbf{det}(\boldsymbol{H(x,y,\sigma)}) = \boldsymbol{\sigma^2}\left(\boldsymbol{L_{xx}(x,y,\sigma)L_{yy}(x,y,\sigma)} - \boldsymbol{L_{xy}}^{\boldsymbol{2}}\boldsymbol{(x,y,\sigma)}\right) \textbf{.(II.12)}$$

Cette étape permet donc détecter les points d'intérêts candidats. L'algorithme comporte ensuite des étapes intermédiaires destinées à apporter plus de précision dans leur localisation.

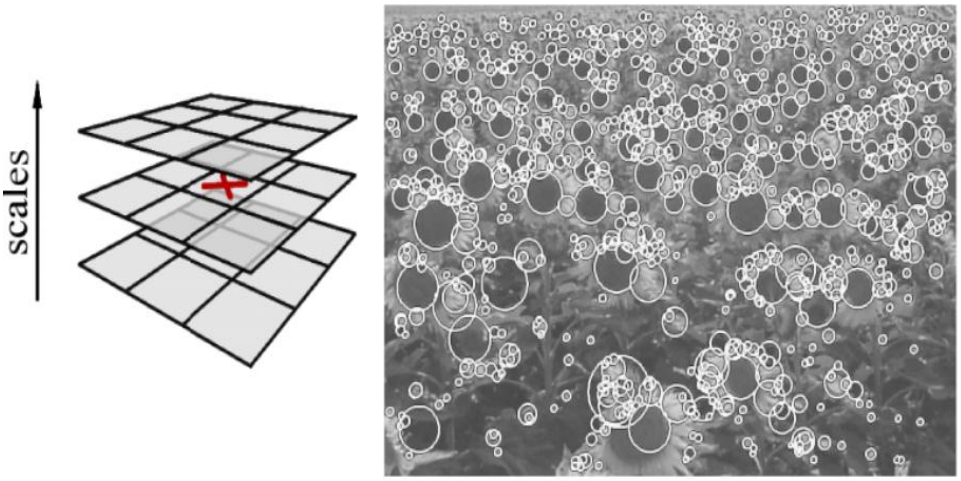


Figure 2.7 : élimination des non-maximum caractéristique (détecteur de caractéristique blob-like)

II.2.2 Descripteur local de l'image

Le calcul des descripteurs se fait grâce aux **ondelettes de Haar**. Elles permettent d'estimer l'orientation locale du gradient et donc d'apporter l'invariance par rapport à la rotation. Les réponses des ondelettes de Haar sont calculées en x et y dans une fenêtre circulaire dont le rayon dépend du facteur d'échelle du point d'intérêt détecté. Ces réponses spécifiques contribuent à la formation du vecteur de caractéristique correspondant au point clé.

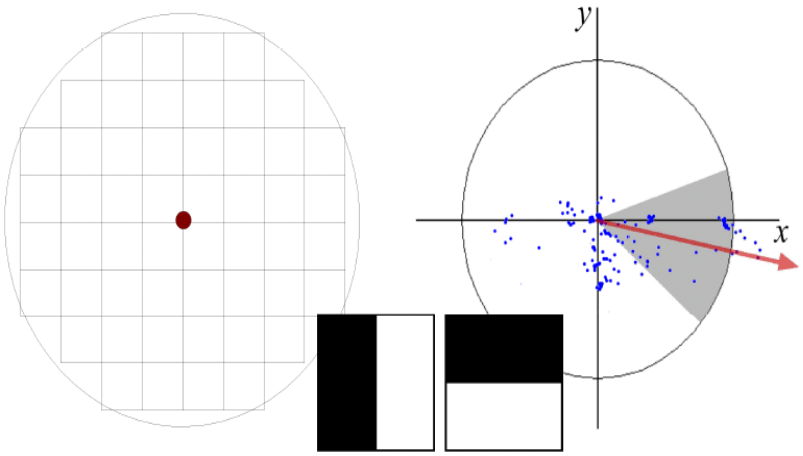


Figure 2.8 : Assignation d'orientation du descripteur.

II.2.3 Appariement des points d'intérêts

Pour ce qui concerne la mise en correspondance des descripteurs, i.e. la recherche de la meilleure similitude entre les descripteurs de deux images, le critère utilisé est le même que celui utilisé dans l'algorithme des SIFT, i.e. celui de la distance euclidienne.

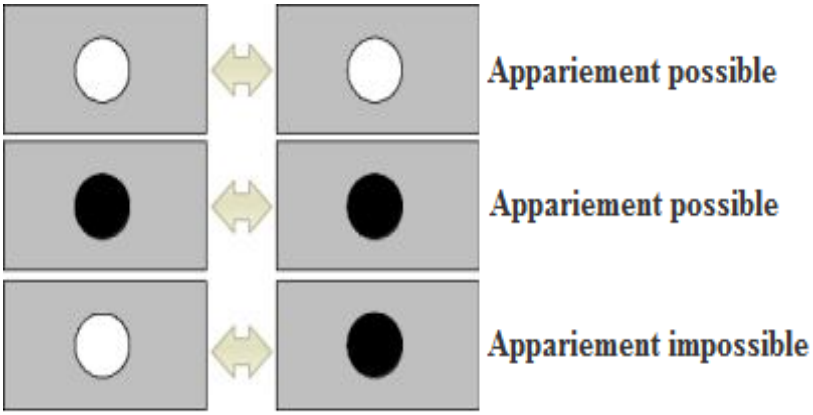


Figure 2.9 : Exemples d'appariements.

II.4 Description de la méthode LBP

La méthode LBP (Local Binary Pattern) a initialement été proposée par ***Ojala*** en 1996 afin de caractériser les textures présentes dans des images en niveaux de gris.elle consiste à attribuer à chaque pixel *P* de l'image *I(i,j)* à analyser, une valeur caractérisant le motif local autour de ce pixel. Ces valeurs sont calculées en comparant le niveau de gris du pixel central *P* aux valeurs des niveaux de gris des pixels voisins [21].

Le concept du LBP est simple, il propose d'assigner un code binaire à un pixel en fonction de son voisinage. Ce code décrivant la texture locale d'une région est calculé par seuillage d'un voisinage avec le niveau de gris du pixel central. Afin de générer un motif binaire, tous les voisins prendront alors une valeur "1" si leur valeur est supérieure ou égale au pixel courant et "0" autrement. Les pixels de ce motif binaire sont alors multipliés par des poids et sommés afin d'obtenir un code LBP du pixel courant.

On obtient donc pour toute l'image, des pixels dont l'intensité se situe entre 0 et 255 comme dans une image à 8 bits ordinaire. Plutôt que de décrire l'image par la séquence des motifs LBP, on peut choisir comme descripteur de texture un histogramme de dimension 255.

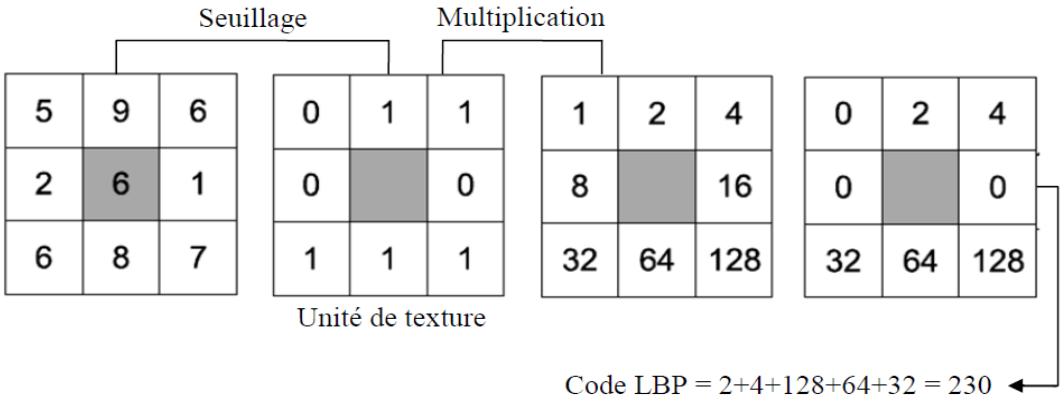


Figure 2.10: Construction d'un motif binaire et calcul du code LBP.

Pour calculer le code LBP dans un voisinage de pixel P, dans un rayon R, on compte simplement les occurrences de niveaux de gris $g_p$plus grands que la valeur centrale.

$$\boldsymbol{LBP_{P;R} = \sum_{p=0}^{p-1} S(g_p - g_c)2^p} \text{(II.13)}$$

Oùg_p et $g_c$sont respectivement les niveaux de gris d'un pixel voisin et du pixel central.

Où s(x) est la fonction signe :

$$\boldsymbol{s(x) = \begin{cases} 1 si x \geq 0 \\ 0 sinon \end{cases}} \text{(II.14)}$$

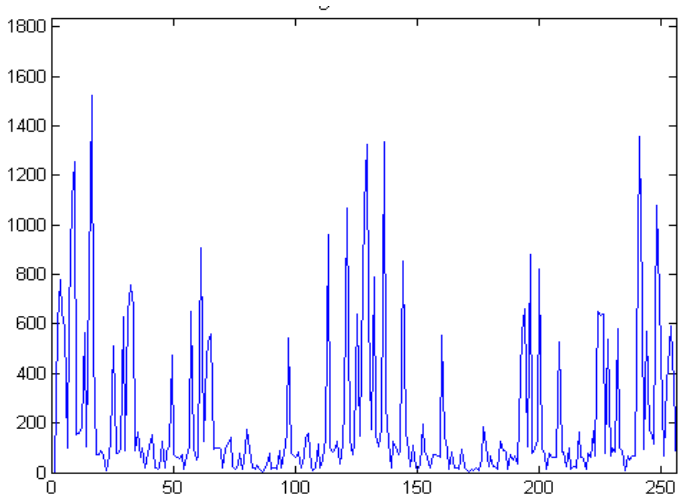


Figure 2.11 : Histogramme finale de la méthode LBP

II.5 Conclusion

Dans ce chapitre, nous avons introduit la description de trois méthodes d'indexation d'image (voire SIFT, LBP et SURF,). Nous avons détaillé les étapes de l'extraction des points d'intérêts et le calcul des descripteurs pour chacune de ces méthodes.

La première méthode SIFT sert à optimiser la structure des données à sauvegarder, en prenant une image et la transformer en vecteurs des descripteurs de plusieurs point considérer comme les informations les plus importants que l'image contient. La deuxième LBP est plus optimisé par rapport à la première en temps d'exécution et le résultat d'information à sauvegarder, mais l'inconvénient majeur de cette méthode est que le type d'information qui n'est pas robuste aux différentes transformations géométrique, tel que la rotation, le changement d'échelle ou d'autre effets lumineux. Troisièmement, la méthode SURF qui est déduit de son prédécesseur la méthode SIFT, les vecteurs résultant sont plus stables, robuste, et même le temps d'exécution de cette méthode est réduit considérablement.

Toutes ces approches ont les mêmes buts à la fin qui sont :

- La réduction de la taille des données a manipulé.
- La réduction du temps du traitement.
- Rendre l'indexation et la recherche du contenu visuel plus flexible.

Comme toutes les approches proposées dans le domaine du traitement d'image, on trouve chaque méthode donne des nouveaux avantages, mais elle n'a pas atteint le niveau demandé par l'utilisateur car chacune d'elles a son taux d'erreur, et elle est encore proposé à être améliorer.

Chapitre 3 :

Clustering et analyse de données

III.1. Introduction

Après avoir traité une image avec une telle méthode et extraire les vecteurs des descripteurs, une phase d'analyse et restructuration de ces données est nécessaire pour avoir une indexation et une recherche presque parfaite en utilisant ces informations extraites, alors pour atteindre ce but une phase intermédiaire entre l'indexation et la recherche doit être conçu pour avoir les meilleures performances d'un tel system.

Durant cette phase intermédiaire, l'analyse prend deux sens indissociables qui sont :

- L'analyse en composants principales(ACP).

- Le Clustering des données.

Il faut toujours éliminer les informations qui ne sont pas nécessaires pour minimiser les coûts de stockage et du temps pris pour gérer ces données, c'est pour ça on se trouve devant une approche nommé analyse en composants principales (ACP). Cette approche qui va être expliqué en détails au sein du chapitre.

La phase de la recherche nécessite une grande simplification des données à gérer, car les approches de recherche sont gourmandes en temps de calcul, vu la masse des descripteurs à rechercher dedans, mais les chercheurs ont proposé la technique du clustering qui prend en charge de partitionner les données en des ensemble plus homogènes pour faciliter la recherche et la diriger au bon sens et rend les résultats plus préciser.

Dans la suite de ce chapitre nous essayerons de présenter en détails les différentes approches telles que l'ACP et le Clustering.

III.2 Clustering

C'est le partitionnement de données (ou data clustering en anglais) est une des méthodes statistiques d'analyse des données. Elle vise à diviser un ensemble de données en différents «paquets» homogènes, en ce sens que les données de chaque sous-ensemble partagent des caractéristiques communes, qui correspondent le plus souvent à des critères de proximité (similarité informatique) que l'on définit en introduisant des mesures et classes de distance entre objets **[22].**

D'une autre part, c'est la collection de méthodes pour regroupement des données non étiquetées en sous-ensembles (appelés Clusters) que l'on croit refléter la structure sous-jacente

des données, basé sur les groups de similarité à l'intérieur des données. Pour cela le Clustering a besoin de conceptualiser les groupes d'où le concept de distance inter-Cluster et intra-Cluster ont été définis, ainsi que dans certains scénarios la définition d'un Cluster Head qui joue le rôle du représentant du Cluster est nécessaire

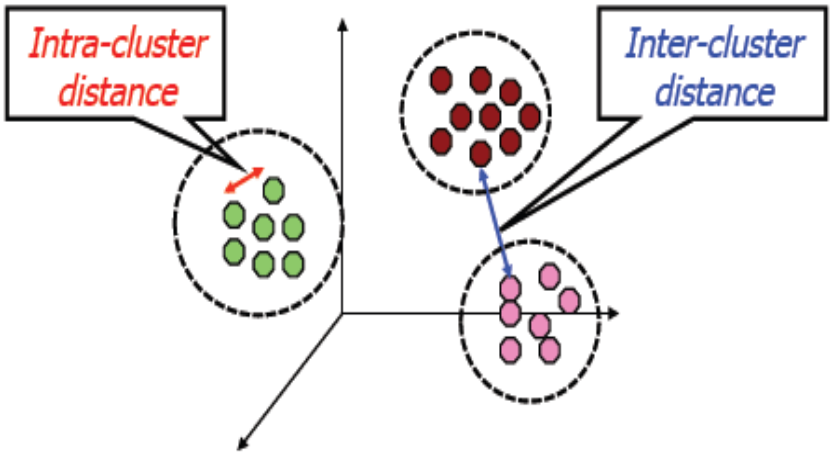


Figure 3.1 : Illustration de la conceptualisation des groupes par distance

III.2.1 L'objectif du Clustering

L'objectif du Clustering est d'atteindre une haute similarité intra-cluster (les objets au sein d'un cluster sont similaires) et une faible similarité inter-cluster (objets de différents cluster sont dissemblables). Il s'agit d'un *critère interne* de la qualité du Clustering. Tout de même les scores d'un critère interne ne traduisent pas nécessairement la performance de l'application. Une alternative à ces critères est l'évaluation directe de l'utilité du Clustering pour l'application, comme mesurer le temps qu'il faut aux utilisateurs de trouver une réponse avec des algorithmes de Clustering différents.

III.2.2 Techniques de Clustering

Il y a un grand nombre d'algorithmes de regroupement et aussi de nombreuses possibilités pour l'évaluation d'un regroupement contre un étalon-or. Le choix d'un algorithme de classification approprié et d'une mesure appropriée pour l'évaluation dépend des objets de regroupement et la tâche de regroupement. Les objets de regroupement au sein de cette thèse sont des vecteurs de descripteurs, et la tâche de classification est une classification de similarité entre ces vecteurs.

En effet, les techniques de clustering se diffèrent, mais ils se trouvent tous sous deux types majeurs qui sont :

- Le clustering hiérarchique.

- Le clustering par partitionnement.

Pour définir mieux les deux types on doit dire que l'hiérarchique se pose sur une architecture arborescente par contre le clustering par partitionnement décompose les individus en k groupe, bien que les deux types ont pour but de simplifier et de rendre la recherche plus facile, flexible et rapide.

III.2.2.1 Clustering hiérarchique

Le concept de base du regroupement hiérarchique est de fusionner successivement les documents en clusters (groupes), puis les clusters entre eux selon leur degré de similarité. Elle se décompose donc en deux étapes répétées en boucle (jusqu'à ce qu'il ne reste qu'un seul cluster unique) :

- Calculer la similarité (distance) entre tous les clusters existant à l'étape en cours ;
- fusionner les deux clusters qui sont les plus similaires.

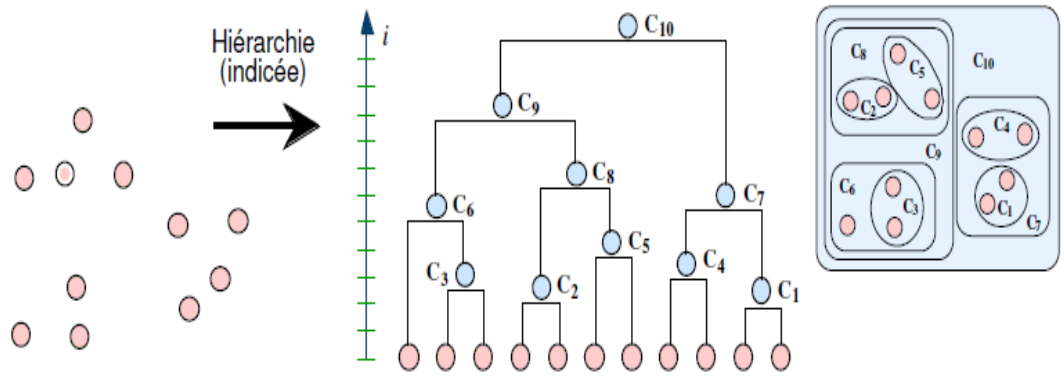


Figure 3.2 : Regroupement et classification selon la méthode hiérarchique

III.2.2.1.1 principe

Initialement, chaque individu forme une classe, soit n classes. On cherche à réduire le nombre de classes à nb_classes< n, ceci se fait itérativement.

À chaque étape, on fusionne deux classes, réduisant ainsi le nombre de classes. Les deux classes choisies pour être fusionnées sont celles qui sont les plus "proches", en d'autres termes, celles dont la dis-similarité entre elles est minimale, cette valeur de dis-similarité est appelée indice d'agrégation. Comme on rassemble d'abord les individus les plus proches, la première itération a un indice d'agrégation faible, mais celui-ci va croître d'itération en itération **[23].**

III.2.2.1.2 Mesure de dis-similarité inter-classe

La dis-similarité de deux classes :

$$C_1 = \{x\}, C_2 = \{y\} \quad \textbf{(III.1)}$$

Contenant chacune un individu se définit simplement par la dis-similarité entre ces individus.

$$dissim(C_1, C_2) = dissim(x, y) \textbf{(III.2)}$$

Lorsque les classes ont plusieurs individus, il existe de multiples critères qui permettent de calculer la dis-similarité. Les plus simples sont les suivants :

- Le saut minimum retient le minimum des distances entre individus de C_1 et C_2 :

$$dissim(C_1, C_2) = \min_{x \in C_1, y \in C_2} \big(dissim(x, y)\big) \textbf{(III.3)}$$

- Le saut maximum est la dis-similarité entre les individus de C_1 et C_2 les plus éloignés :

$$dissim(C_1, C_2) = \max_{x \in C_1, y \in C_2} \big(dissim(x, y)\big) \textbf{(III.4)}$$

- Le lien moyen consiste à calculer la moyenne des distances entre les individus de C_1 et C_2:

$$dissim(C_1, C_2) = moyenne_{x \in C_1, y \in C_2} \big(dissim(x, y)\big) \textbf{(III.5)}$$

- La **distance de Ward** vise à maximiser l'inertie inter classe :

$$dissim(C_1, C_2) = \frac{n_1 * n_2}{n_1 + n_2} dissim(G_1, G_2) \textbf{(III.6)}$$

- avec n_1 et n_2 les effectifs des deux classes G_1 et G_2

III.2.2.2 Clustering par partitionnement

Le clustering par partitionnement est une approche qui permet de subdiviser l'ensemble des individus en un certain nombre de classes en employant une stratégie d'optimisation itérative, dont le principe général est de générer une partition initiale, puis de chercher à l'améliorer en réattribuant les données d'une classe à l'autre.

Ces algorithmes recherchent donc des maxima locaux en optimisant une fonction objectif traduisant le fait que les individus doivent être similaires au sein d'une même classe, et dissimilaires d'une classe à une autre **[24]**. Les classes de partition final, prises deux à deux, sont d'intersection vide est représentée par noyau.

Principe

Comme il est déjà vu les algorithmes de clustering par partitionnement sont itératifs, on peut déduire trois étapes majeures pour ces algorithmes :

- Initialement, c'est la création de K-groupe des clusters ou les données vont être groupées, en donnant les centroids et initialisé les individus qui vont être traité.

- Durant les itérations prédéfinies l'algorithme cherche à quel cluster il assigne l'individu testé.

- Finalement, l'algorithme s'arrête quand il y'on pas des changements dans les clusters c.-à-d. que l'intersection entre les clusters donne l'ensemble vide.

III.2.2.2.1 K-means

Le partitionnement en k-moyennes (ou k-means en anglais) est une méthode de partitionnement de données, c'est le plus célèbre algorithme de Clustering par partitionnement. Son objectif est de minimiser la moyenne des distances carrées entre les objets et le centre du cluster.

Étant donnés des points et un entier k, le problème est de diviser les points en k partitions, souvent appelés clusters, de façon à minimiser une certaine fonction. On considère la distance d'un point à la moyenne des points de son cluster, la fonction à minimiser est la somme des carrés de ces distances.

Il existe une heuristique classique pour ce problème, souvent appelée méthodes des k-moyennes, utilisée pour la plupart des applications. Les k-moyennes sont notamment utilisées en apprentissage non supervisé où l'on divise des observations en K-partitions. Les nuées dynamiques sont une généralisation de ce principe, pour laquelle chaque partition est représentée par un noyau pouvant être plus complexe qu'une moyenne. L'algorithme classique de K-means est le même que l'algorithme de quantification de Lloyd-Max.

Principe

Vu l'inexistence d'un standard défini pour l'algorithme de K-means, les développeurs utilise cet algorithme selon ces besoins c'est pour cela on trouve beaucoup d'implémentation et définitions différentes dont le but est unique, dans ce qui suit on note quelques définition du principe du K-means :

Étant donné un ensemble de points ($\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$), on cherche à partitionner les n points en k ensembles $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ ($k \leq n$) en minimisant la distance entre les points à l'intérieur de chaque partition :

$$arg_S min \sum_{i=1}^{k} \sum_{x_j \in s_i} \|x_j - \mu_i\|^2 \textbf{(III.7)}$$

Où μ_i est la moyenne des points dans S_i.

Aussi McQueen a défini un des plus simples algorithmes de classification automatique des données, où l'idée principale est de choisir aléatoirement un ensemble de centres fixé a priori et de chercher itérativement la partition optimale**.**

Chaque individu est affecté au centre le plus proche, après l'affectation de toutes les données la moyenne de chaque groupe est calculé, elle constitue les nouveaux représentants des groupes, lorsqu'ont abouti à un état stationnaire (aucune donnée ne change de groupe) l'algorithme est arrêté **[25].**

L'algorithme K-Means fonctionne comme suit :

- Choisir k moyennes $\mathbf{m}1^{(1)}, \ldots, \mathbf{m}k^{(1)}$ initiales
- Répéter jusqu'à convergence :

* assigner chaque observation à la moyenne la plus proche (*i.e* effectuer une partition de Voronoï selon les moyennes).

$$\mathbf{S_i^{(t)}} = \left\{\mathbf{x_j} : \left\|\mathbf{x_j} - \mathbf{m_i^{(t)}}\right\| \leq \left\|\mathbf{x_j} - \mathbf{m_{i*}^{(t)}}\right\| \textbf{ pour tous } \mathbf{i^* = 1, \dots, k}\right\}$$

* mettre à jour la moyenne de chaque cluster

$$m_i^{(t+1)} = \frac{1}{\left[S_i^{(t)}\right]} \sum_{x_j \in S_i^{(t)}} x_j \textbf{(III.8)}$$

* La convergence est atteinte quand il n'y a plus de changement.

III.2.2.2.2 Bisecting K-means

Bisecting K-means est une technique de classification et une combinaison entre la classification par partitionnement et la classification hiérarchique, il se base sur les mêmes principes du K-means d'un point de vue technique. Il est une technique très puissante pour la réduction de la taille des vecteurs de caractéristique utilisés dans la classification. Les vecteurs similaires l'un à l'autre vont être groupé dans le même cluster.

Principe :

Cet algorithme comporte comme le k-means avec quelques modifications, et dans ce qui suit on cite les étapes de l'algorithme :

(1) Un cluster primaire C_0 qui contient l'ensemble d'individus$S\{i_1, i_2, i_3, \dots \dots \dots, i_n\}$ et initialisation de nombre des clusters demandé à la fin de l'algorithme.

(2) On applique l'algorithme basic du K-means pour obtenir deux clusters C_1 et C_2 , et on les intègre comme des feuilles du cluster C_0

(3) Calculer les distances Dist1 et Dist2 pour les clusters résultant de l'étape précédente et comparer les, si Dist1>Dist2 on prend le cluster C_1 et appliquer l'algorithme de K-means et intégrer les deux nouveau clusters C_3et C_4 comme des feuilles du cluster C_1.

(4) On répète l'étape 3 jusqu'à ce qu'on atteint le nombre des clusters demandé au début.

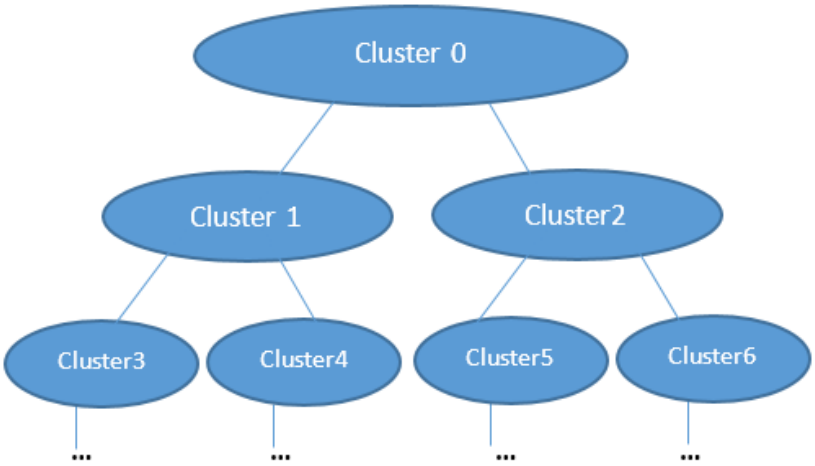


Figure 3.3 : Exemple de résultats obtenu par l'algorithme Bisecting K-means

III.3 Analyse de données

L'analyse des données est une famille de méthodes statistiques dont les principales caractéristiques sont d'être multidimensionnelles et descriptives. Certaines méthodes, pour la plupart géométriques, aident à faire ressortir les relations pouvant exister entre les différentes données et à en tirer une information statistique qui permet de décrire de façon plus succincte les principales informations contenues dans ces données. D'autres techniques permettent de regrouper les données de façon à faire apparaître clairement ce qui les rend homogènes, et ainsi mieux les connaître.

L'analyse des données permet de traiter un nombre très important de données et de dégager les aspects les plus intéressants de la structure de celles-ci. Le succès de cette discipline dans les dernières années est dû, dans une large mesure, aux représentations graphiques fournies. Ces graphiques peuvent mettre en évidence des relations difficilement saisies par l'analyse directe des données ; mais surtout, ces représentations ne sont pas liées à une opinion « a priori » sur les lois des phénomènes analysés contrairement aux méthodes de la statistique classique **[26].**

Les fondements mathématiques de l'analyse des données ont commencé à se développer au début du XXe siècle, mais ce sont les ordinateurs qui ont rendu cette discipline opérationnelle, et qui en ont permis une utilisation très étendue.

III.3.1 analyse par réduction des dimensions

La représentation des données multidimensionnelles dans un espace à dimension réduite est le domaine des analyses factorielles, analyse factorielle des correspondances, analyse en composantes principales, analyse des correspondances multiples. Ces méthodes permettent de représenter le nuage de points à analyser dans un plan ou dans un espace à trois dimensions, sans trop de perte d'information, et sans hypothèse statistique préalable. En mathématiques, elles exploitent le calcul matriciel et l'analyse des vecteurs et des valeurs propres.

III.3.2 Analyse en composantes principales (ACP)

L'Analyse en Composantes Principales (ACP) est l'une des méthodes d'analyse de données multi-variées les plus utilisées. Dès lors que l'on dispose d'un tableau de données quantitatives (continues ou discrètes) dans lequel n observations (des individus, des produits, …) sont décrites par p variables (des descripteurs, attributs, mesures, …), si p est assez élevé, il est impossible d'appréhender la structure des données et la proximité entre les observations en se contentant d'analyser des statistiques descriptives uni-variées ou même une matrice de corrélation **[27].**

III.3.2.1 Utilisations de l'Analyse en Composantes Principales

Il existe plusieurs applications pour l'Analyse en Composantes Principales, parmi lesquelles :

- l'étude et la visualisation des corrélations entre les variables, afin d'éventuellement limiter le nombre de variables à mesurer par la suite

- l'obtention de facteurs non corrélés qui sont des combinaisons linéaires des variables de départ, afin d'utiliser ces facteurs dans des méthodes de modélisation telles que la régression linéaire, la régression logistique ou l'analyse discriminante.

- la visualisation des observations dans un espace à deux ou trois dimensions, afin d'identifier des groupes homogènes d'observations, ou au contraire des observations atypiques.

III.3.2.2 Principe de l'Analyse en Composantes Principales

Les différentes opérations de l'ACP sont :

1. Le calcul des matrices de covariance et de corrélation de l'image couleur, ce sont des matrices carrées dont la dimension est égale à la dimension de l'espace d'image.

Soit $X_{n\times p}$ une matrice contenant les données :

$$X = \begin{Bmatrix} x_{11} \dots \dots x_{1p} \\ x_{21} \dots \dots x_{2p} \\ \dots \dots \dots \dots \\ x_{n1} \dots \dots x_{np} \end{Bmatrix}$$

Où x_{ij} est la valeur de l'individu i pour la variable j, j représente les coordonnées du pixel dans l'espaces utilisé.

La matrice R des corrélations est définie ainsi :

$$R = \begin{Bmatrix} 1 r_{21} \dots r_{1p} \\ r_{21} 1 \dots r_{2p} \\ \dots \dots \dots \dots \\ r_{P1} r_{P2} \dots 1 \end{Bmatrix}$$

Où $r_{ij} = corr(Xi; Xj)$ est le coefficient de corrélation entre les variables Xi et Xj.

2. Le calcul des valeurs et vecteurs propres : la matrice R des corrélations est une matrice symétrique. Elle peut être écrite comme suit : $R = U \wedge U'$

Où ∧est la matrice des valeurs propres ordonnées telles que

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$$

Avec :

$$\lambda = \begin{Bmatrix} \lambda 0 \dots 0 \\ 0 \lambda \dots 0 \\ \dots \dots \dots \dots \\ 0 0 \dots \lambda \end{Bmatrix}$$

Et $U(p \times p) = (u1|u2| \dots \vee up)$, est la matrice des vecteurs propres associés aux valeurs propres.

3. Le calcul des composantes principales de l'image couleur.

Après le calcul des valeurs propres noté λ_i et d'extraire les vecteurs propres correspondants notés$u1$.

Le calcul des nouvelles composantes Yi se fait donc par la relation :

$$Yi = uiX_i^T$$

III.4 Conclusion

Ce chapitre met l'accent sur le besoin d'un accès rapide et précis à l'information dans le cas des systèmes de recherche par contenu visuel, notamment la nécessité d'une une organisation des descripteurs du contenu qui tire profit de la relation entre leurs données. Cela à influence directement sur la qualité de la recherche et la pertinence des résultats retournés.

Après avoir établi un bref état de l'art sur les approches de Clustering, nous avons vu la technique d'analyse en composantes principale qui transfère et restructure les données retournées de la phase d'indexation et les rendre plus claire et optimisé en sortant les informations les plus importants d'un ensemble de données.

Chapitre 4 :

Description générale du système proposé

IV.1 Introduction

Notre travail se place dans le contexte de la recherche d'image par contenu visuel (CBIR), nous avons développé un système générique indépendant d'un domaine d'application particulier ce qui justifie le choix des base d'images hétérogènes non spécifiques.

Au sein de ce chapitre, nous avons présenté le système proposé d'une façon détaillé, commençons par son architecture générale avec ces deux phases, la phase d'indexation étape par étape, ensuite la phase de classification « Clustering » et la recherche, enfin le calcule de similarité et l'affichage des résultats pour l'utilisateur de notre système.

IV.2 Architecture du système

Le système proposé se base principalement sur deux phases générales :

- **La première phase est la phase d'indexation :**

Cette phase s'occupe de l'extraction des descripteurs de la méthode SIFT et la méthode LBP et l'indexation des informations qui ont été extraites d'après une image ou une base d'images complète.

- **La deuxième phase est la recherche par contenu :**

Dans cette phase l'utilisateur va donner une image requête, dont laquelle le système commence par l'extraction des descripteurs des méthodes SIFT et/ou la méthode LBP de l'image requête, puis il récupère toutes les images similaires à notre requête et enfin il calcule le taux de similarité entre eux.

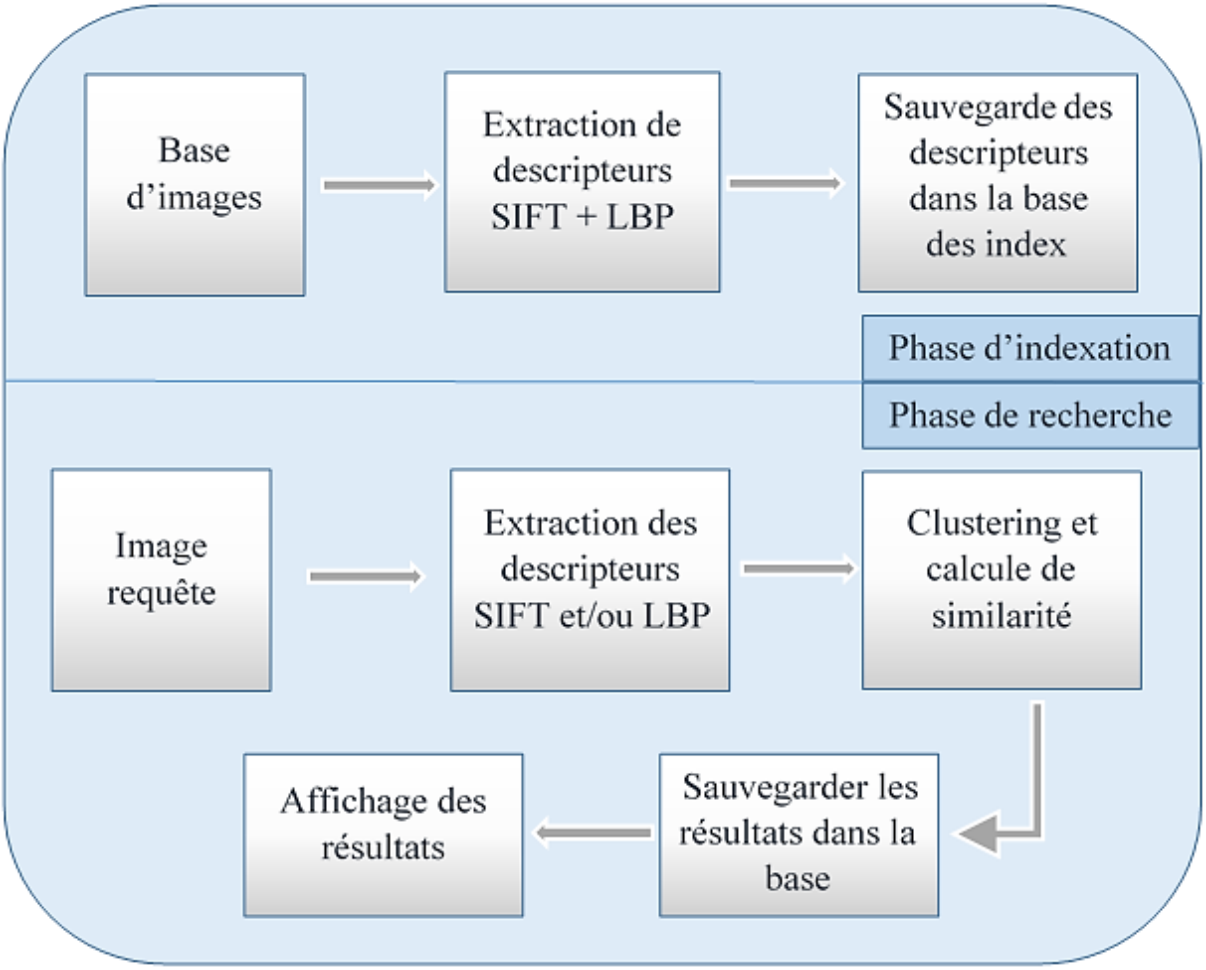


Figure 4.1 : Schéma générale du système proposé

IV.2.1 Phase d'indexation

La phase d'indexation ce déroule en deux étapes principales :

- l'extraction des informations
- la sauvegarde des informations dans la base des indexes

Dans cette phase, nous avons utilisé deux méthodes « SIFT » et/ou « LBP » pour indexer notre base d'images.

Dans un premier temps, nous avons implémenté L'algorithme de SIFT de David G.lowe est comme suit :

IV.2.1.1 Extraction des points d'intérêt

Pour extraire ces points le système doit suivre un ordre d'étapes pour accomplir cette tâche :

- Conversion l'image de son espace couleur au niveau de gris.
- Construction de la pyramide des images : la pyramide obtenue est composé des cinq octaves d'images ou chacune des octaves se compose de cinq images dont la taille est le quart de l'octave précédent.

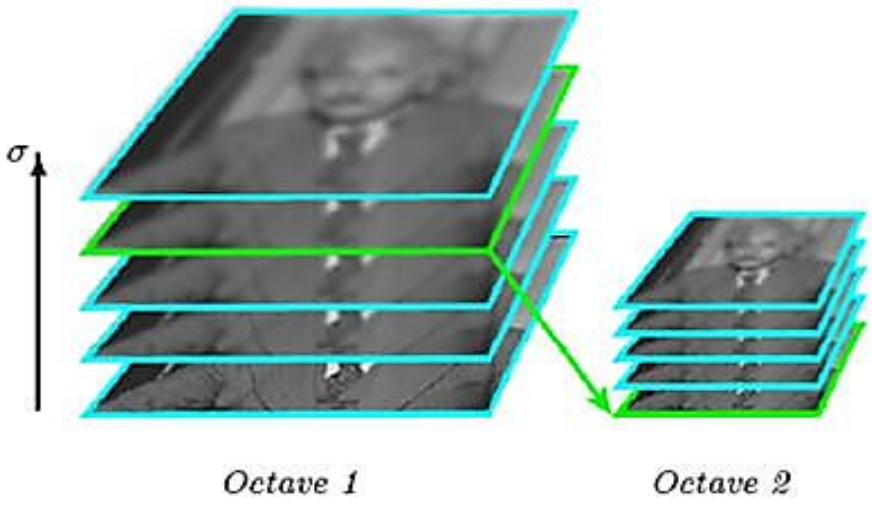


Figure 4.2 : Exemple d'octaves d'images crée.

- Application du filtre gaussien pour chaque image avec différentes valeurs de **σ**, puis nous avons calculé la différence entre chaque deux octave.

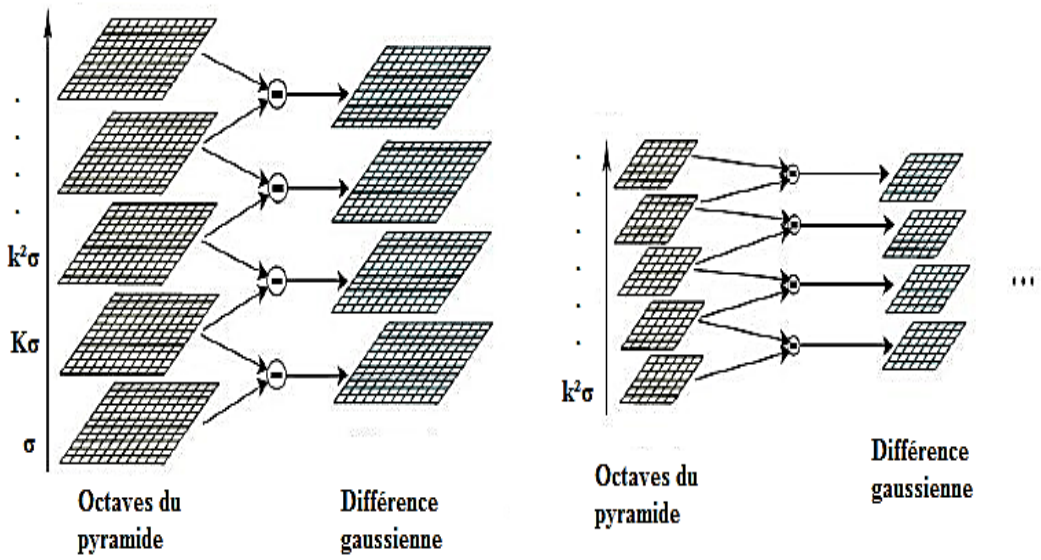


Figure 4.3 : application du filtre gaussien et calcule les différences

- L'extraction de tous les points d'intérêt possible dans les

Quadripôles résultants avec la condition qu'il soit le plus petit ou

Le plus grand de ces 26 voisins.

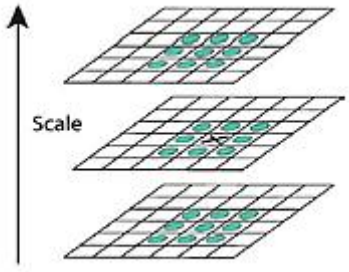


Figure 4.4 : extraction de tous les points d'intérêt possible

IV.2.1.2 Filtrage

Après avoir trouvé tous les points d'intérêt possible, une étape de filtrage est nécessaire pour améliorer la précision des points d'intérêt, nous avons éliminé les points qui situent sur des contours ou les points qui ont un faible contraste.

Figure 4.5 : élimination des points d'intérêt de faible contraste et qui se trouvent sur des contours.

IV.2.1.3 Calcule des descripteurs

L'étape finale nous avons calculé les descripteurs des points d'intérêt obtenus dans l'étape de filtrage, en utilisant les étapes suivantes :

- Le calcule de gradient orienté des voisins du point d'intérêt dont le bloc de calcule est de taille 16x16 dans le centre c'est un le point d'intérêt.

- La construction de l'histogramme du gradient orienté de cette fenêtre de chaque bloc de taille 4x4 du bloc global 16x16.

- La transformation des résultats du bloc global en vecteur de 128 valeurs.

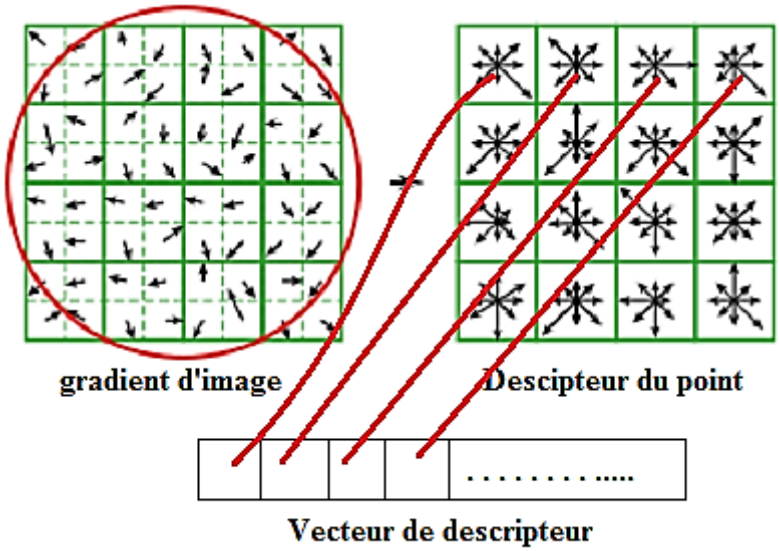


Figure 4.6 : calcule de descripteur

Ensuite, nous avons implémenté L'algorithme de LBP comme suit :

IV.2.2 L'histogramme LBP

Dans cette approche LBP « Local Binary Pattern », nous avons construit un histogramme depuis les valeurs calculées à partir des huit voisins de chaque pixel, l'algorithme LBP se déroule comme suit :

- Le parcourt l'image pixel par pixel.
- Pour chaque pixel nous avons calculé la différence entre le pixel et ses huit voisins.
- Si la valeur du voisin est inférieur à celle du pixel alors on prend 0 si non 1.
- La construction de l'histogramme à partir des motifs obtenu par la conversion du code binaire en code décimale.

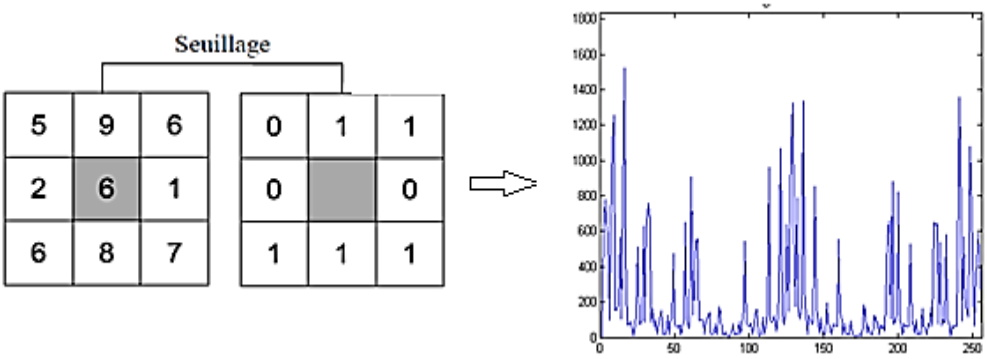


Figure 4.7 : Construction d'histogramme LBP

IV.2.3 Phase de recherche et d'appariement

Pour accomplir cette phase on est besoin d'une image exemple donné par l'utilisateur et une base d'image indexée, aussi cette phase passe par plusieurs étapes avant de rendre les résultats :

IV.2.3.1 Indexation de l'image requête

Le système proposé donne à l'utilisateur le choix d'utilisé différentes méthodes pour la recherche soit en utilisant la méthode SIFT ou LBP ou la combinaison des deux méthodes.

Quand la phase de recherche commence, la première étape est d'indexer l'image requête au premier lieu pour faire extraire les vecteurs des descripteurs et les informations qui vont être comparé au celles de la base d'images.

- Si l'utilisateur utilise la méthode SIFT, le système n'indexe qu'avec les descripteurs SIFT.

- Si l'utilisateur utilise la méthode LBP, le système n'indexe que avec l'histogramme LBP.

- Si l'utilisateur utilise la méthode SIFT et LBP le système indexe que les deux type de descripteurs SIFT et LBP

IV.2.3.2 Processus de la recherche

Dans cette étape, notre système récupère les informations indexé à la base d'index pour les faire comparer aux descripteurs de l'image requête.

Vu la masse d'informations à traiter, pour optimiser le temps du traitement une étape de classification « Clustering » est mise au point pour diminué le temps de recherche partiel et global, c'est pour ça notre système utilise **K-means** algorithme pour accomplir cette tâche.

IV.2.3.2.1 Etape de Clustering

Le Clustering est la première étape ou le système crée les classes dédiées à contenir et classifier les descripteurs des images récupérées de la base des indexes, chaque classe crée est centrée sur un descripteur de l'image requête.

Les points vont être assignés à leur cluster le plus proche en calculant la distance entre eux et les centre des clusters.

Le calcule de distance ce fait pour chaque vecteur de descripteur avec tous les centre des clusters, en utilisant la distance euclidienne dans l'espace 128 dimensions puis choisir la distance minimale et assigner le point au cluster jusqu'au l'assignation de tous les points.

IV.2.3.2.2 Calcule de similarité

Après la phase de Clustering, nous avons calculé la moyenne de similarité entre chaque centre de cluster et le descripteur de requête, en utilisant la corrélation. En effet, nous avons calculé la moyenne des corrélations de tous les clusters pour trouver le taux final de similarité.

IV.2.3.2.3 Calcule de similarité entre histogrammes LBP

Si l'utilisateur choisis la méthode LBP comme moyen de recherche, le système récupère les histogrammes indexés dans la base et calcule l'histogramme de l'image requête, puis il fait la corrélation entre chaque histogramme dans la base des indexes et l'histogramme de l'image requête en utilisant la corrélation comme mesure de similarité.

IV.2.3.2.4 indexation des résultats de recherche

Après avoir calculé la similarité entre les images de la base et l'image requête, le système sauvegarde temporairement ces résultats dans la base pour être facile à les récupérer dans le cas où l'utilisateur change le niveau de précision des résultats demandé.

IV.2.3.2.5 affichage des résultats de recherche

Enfin, notre système va récupérer les images de la base et les affichent selon un ordre décroissant où les résultats affichés ont une valeur de similarité supérieure ou égale à celle que l'utilisateur a demandée.

IV.3 Conclusion

Dans ce chapitre nous avons décrit notre système de façon générale en définissant les principales phases du traitement qui lui sont attribuées. Nous avons spécifié le comportement de notre système, le processus et les types d'indexation, le processus de recherche caractérisé par l'exploitation les classes construites par K-means.

Dans le prochain chapitre, nous passerons à l'implémentation du système, nous commencerons par spécifier l'environnement et le contexte de l'implémentation. Enfin nous passons à l'évaluation des performances de notre système à travers quelques tests et expérimentations.

Chapitre 5 :

Implémentation et évaluations

V.1 Introduction

Après avoir donné la description générale du système de recherche d'image par contenu visuel que nous avons développé, nous décrivons dans ce chapitre l'organisation logicielle de ses trois modules à savoir module Indexation, module Clustering et enfin module Recherche, qui ont pour fin, satisfaire les contraintes d'un système de recherche d'information, et assurer une pertinence des résultats retournés à l'utilisateur.

D'un point de vu générale, ce chapitre est composé de deux parties qui englobent le fonctionnement du système proposé qui sont : la phase d'indexation avec ces modules et la phase de recherche du Clustering au calcul des résultats, puis la partie teste et validation qui vas nous donne les résultats que le système a rendu avec les différents tests, en utilisant trois bases d'images de tests **(Z.Wang, Coil, Fei Fei li)**.

V.2 Implémentation

V.2.1 Contexte et environnement

Nous avons utilisé le langage Java pour le développement de notre système sous NetBeans IDE dédié pour le langage Java.

Nous avons choisi ce langage à cause de ces bénéfices tel que :

- la simplicité d'appliqué des traitements sur les images,

- ça flexibilité dans la manipulation des bases de donné relationnel avec la bibliothèque SQLite.

V.2.2 Module d'indexation

Ce module se compose de vingt-deux méthodes où chacune à son tour, et elle fait une tâche partielle de l'étape d'extraction des descripteurs depuis une image passé par le constructeur de la classe « Process », dans ce qui suit nous avons présenté la liste de ces méthodes les plus importantes avec des brefs définitions de ces rôles.

Ce module contient les fonctions qui calculent le descripteur de SIFT et LBP:

1- Les fonctions de SIFT :

```
public process (BufferedImage image,String path,String Name);
```

Constructeur de la classe.

```
public int[][] lapofgauss(int [][] img1,int [][] img2);
```

Rend la différence gaussien entre deux images

```
public static int[][] grayScal(BufferedImage img);
```

Rend une matrice 2D

```
public void scalImage();
```

Construit le pyramide des images retaillé

```
public Vector<int []> keyDetect(int[][],imgB,int[][]imgC,int[][] imgA);
```

Extraction des points intérêts

```
public double[][][] kernelCreator(double sigma,double lvl,int size);
```

Crée le noyau du filtre gaussien

```
public int[][][] gaussianBlur(int[][][] grayArray,double[][][] kernel);
```

Applique le filtre gaussien

```
public void descriptorCalculator(int[] pts);
```

Calcule les orientations et les magnitudes du descripteur

```
public Vector edgedPtsElimanation(Vector<int[]> vect);
```

Enlève les point qui situe sur des coins

```
public Vector LowContrastPtsElimanation(Vector<int[]> vect);
```

Enlève les points qui ont du bas contraste

```
public void index(String path,String name,String siftDis,String lbpDis);
```

Sauvegarde les résultats d'extraction dans la base des indexes.

2- Les fonctions de LBP :

```
public void lbp();
```

Calcule le descripteur de l'approche LBP

V.2.3 Module de Clustering

Pour le clustering nous avons appliqué l'algorithme de classification non supervisé nommé K-means, il se compose de trois classes principales qui sont : Kmeans, Point, Cluster, où chaque classe joue un rôle indissociable, mais la classe la plus importante est Kmeans qui englobe les méthodes qu'on vas les lister avec ces définitions.

```
public Kmeans(String str1,String str2);
```

Constructeur de la classe

```
public void iteration();
```

itération de l'algorithme Kmeans s'occupe de la classification des points a leur classes

```
Public void chargerData(String str1,String str2);
```

Crée les clusters et les points à être classé

```
public double[][][] Discriptors(String str);
```

Décode la chaine de caractères du descripteur

```
public double correlation();
```

calcule la similarité entre les centre des clusters et les point associé

```
public double correlationHistograme(String str,String str2);
```

Calcule la similarité entre deux histogrammes LBP.

V.2.3 Module de recherche

La recherche dans le système proposé se fait à partir de la combinaison des modules d'indexation et clustering, où le système utilise le premier pour indexer l'image requête, puis il fait appel au module de clustering pour définir l'association des point d'intérêts à leurs clusters, puis il calcule la similarité entre les images de la base avec l'image requête.

Tous ces appels de fonctionnement se font à partir de la classe protocole qui s'occupe des protocoles d'indexation et de recherche.

V.3 Fonctionnement

Dans cette partie nous avons présenté le système proposé avec des exemples visuels pris durant son fonctionnement. Où les étapes sont :

V.3.1 Présentation de l'application

L'interface graphique de l'application est très simple et accessible pour l'utilisateur, elle contient deux panneaux (outils et affichage des résultats).

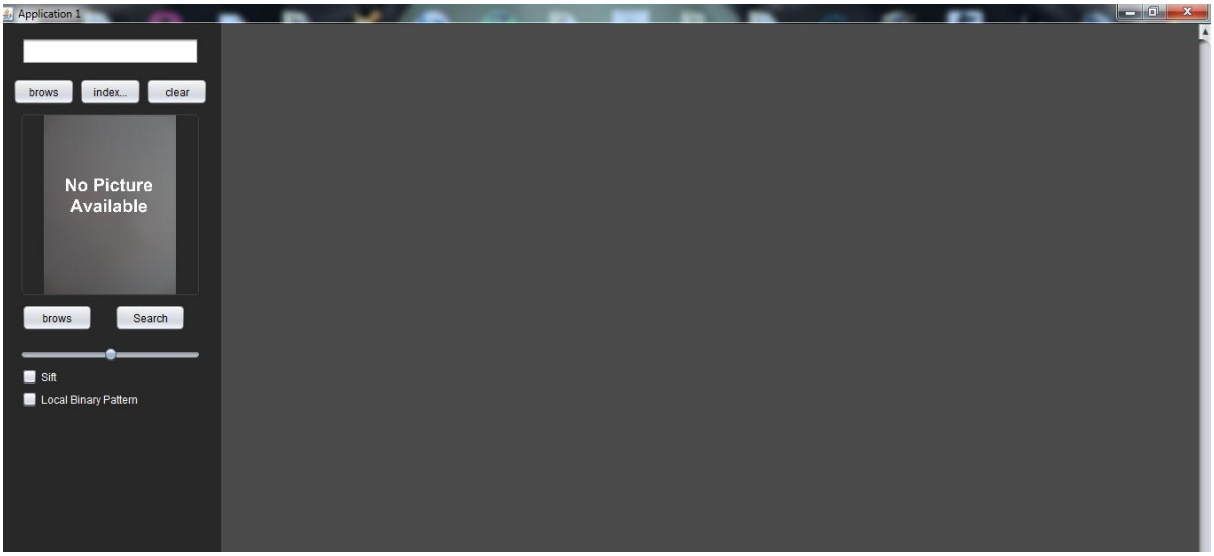


Figure 5.1 : Présentation de l'interface graphique du système proposé

V.3.2 Processus d'indexation

Le processus ce fait par choisir un fichier qui contient des images « une banque d'images » puis on clique sur indexDB.

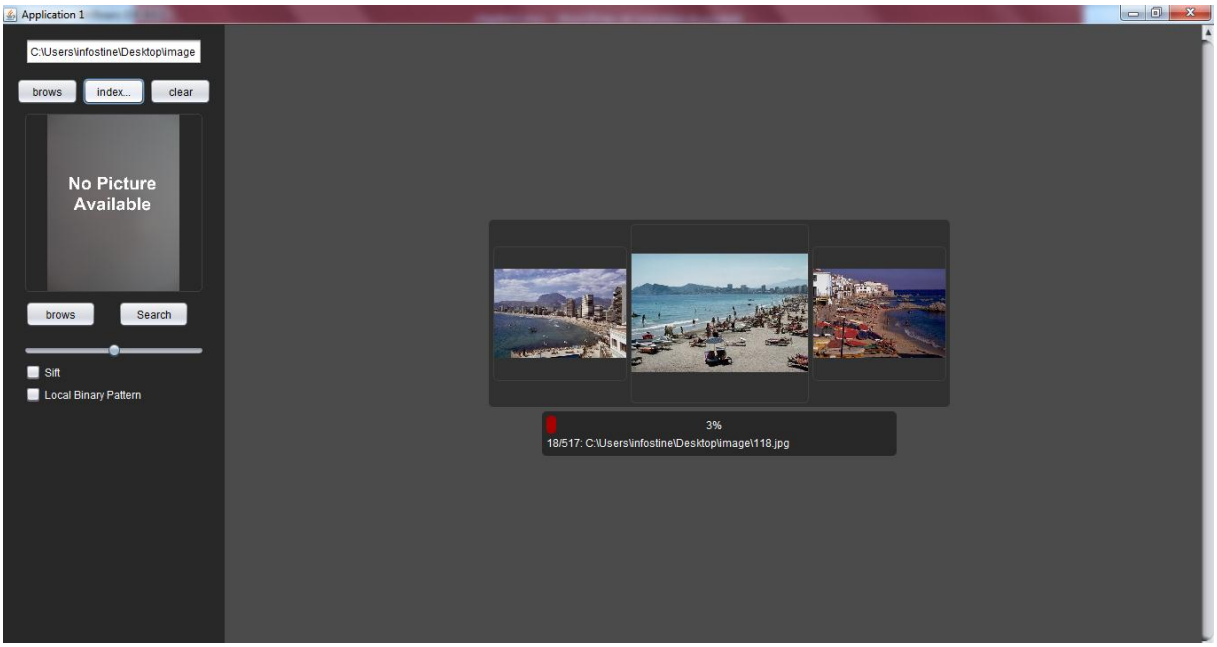


Figure 5.2 : le système durant le processus d'indexation

V3.3 Processus de recherche

Le processus de recherche se fait par choisir une image requête et la méthode de recherche soit en utilisant SIFT soit LBP soit les deux à la fois, le clustering se fait en arrière plans, pour cacher la complexité du système pour les utilisateurs simple.

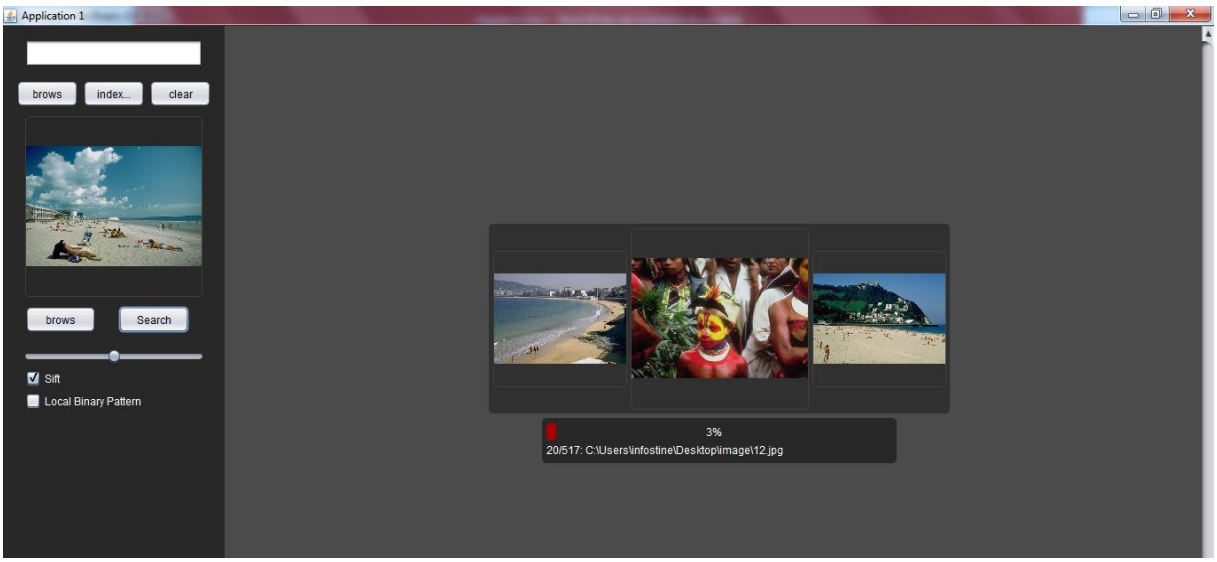


Figure 5.3 : le système durant le processus de recherche

Après avoir finir la phase de recherche le système affiche les résultats où le taux de similarité est supérieur ou égal à la valeur choisi par un JSlider de 1% à 100 %.

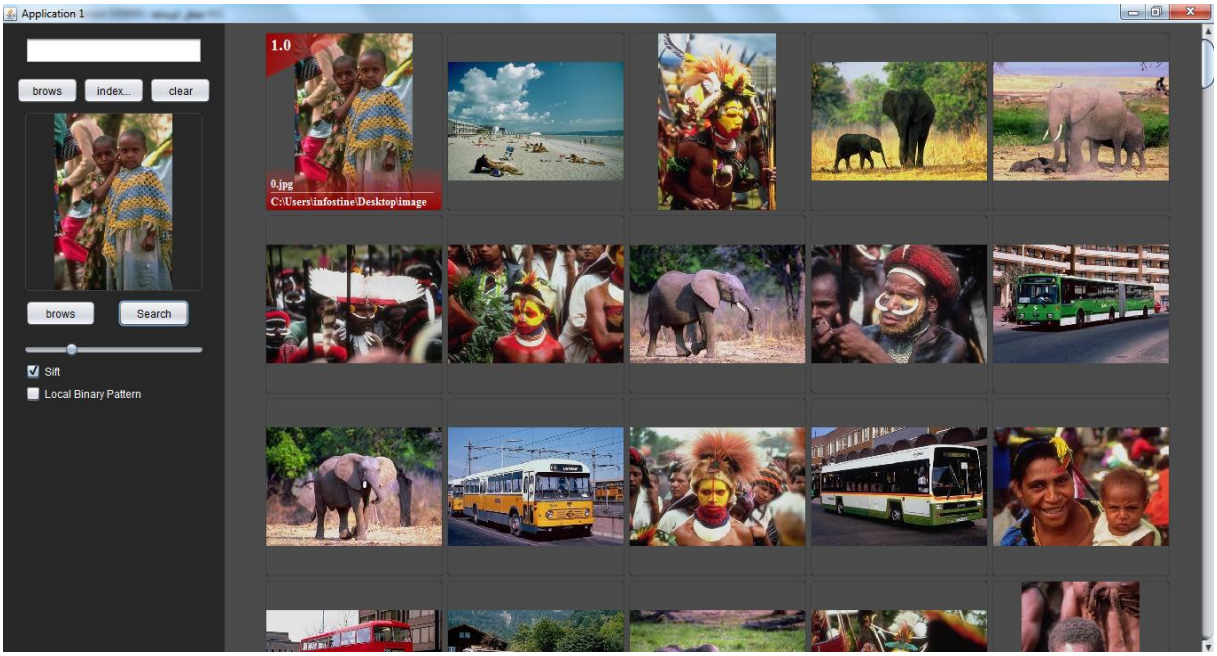


Figure 5.4 : quelques résultats de la recherche en utilisant une image exemple

V.4 Test et validations

Dans cette partie on va présenter les résultats et les graphes des tests d'indexation et recherche sur des différentes bases d'images utilisé.

V.4.1 plateforme de test

La plateforme utilisée pour ce projet est celle de **Microsoft Windows 7** avec **l'IDE NetBeans** pour le langage **Java** et **JRE** « **J**ava **R**untime **E**nvironment » 1.8.

V.4.2 Bases d'images utilisées

Pour évaluer et valider notre système, nous avons utilisé trois bases d'images. Ces bases d'images sont disponibles sur Internet librement. Ces bases d'images possèdent déjà des classes définies où chaque image n'appartient qu'à une seule classe.

V.4.2.1 La base de Wang

La base d'images de Wang est un sous-ensemble de la base d'images Corel. Cette base d'images contient 1000 images naturelles en couleurs. Ces images ont été divisées en 10 classes, chaque classe contient 100 images. L'avantage de cette base est de pouvoir évaluer les résultats. Cette base d'images a été utilisée pour faire des expériences de classification.

Cette base d'images a été créée par le groupe du professeur **Wang** de l'université Pennsylvania State et est disponible à l'adresse : http://wang.ist.psu.edu/.

Chaque image dans cette base d'images a une taille de 384 × 256 pixels ou 256 × 384 pixels**[27].**

Figure 5.5 : 10 classes de la base de **Wang** (**Deselaers**, 2003)

V.4.2.2 Coil (Columbia Object Image Library)

Cette base d'images est très connue pour la reconnaissance des objets. Il y a deux bases d'images **COIL** : **COIL**-20 qui contient des images en niveaux de gris prises à partir de 20 objets différents et **COIL**-100 qui contient des images en couleurs prises à partir de 100 objets différents. Les deux bases d'images consistent en des images prises à partir des objets 3D avec des positions différentes. La base **COIL**-100 a 7200 images en couleurs (100 objets x 72 images/objet). Chaque image a une taille de 128×128 pixels. La base COIL-20 a 1440 images en niveaux de gris (20 objets x 72 images/objet). Chaque image a taille 128×128 pixels **[28].**

Ces bases d'images sont disponibles à l'adresse :

http://www1.cs.columbia.edu/CAVE/research/softlib/

Figure 5.6 : Les objets utilisés dans **COIL**-100(Deselaers, 2003)

Figure 5.7 : Les objets utilisés dans **COIL**-20 (Deselaers, 2003)

V.4.2.3 Base de Fei Fei

Cette base contient des images de 101 objets collectées par Fei-Fei Li, Marco **Adreetto** et **Marc Aurolio Ranzato**. Avec chaque objet, de 40 à 800 images ont été prises. Chaque image a une taille de 300×200 pixels **[29]**. Ces images sont disponibles à l'adresse :

http://www.vision.caltech.edu/feifeili/Datasets.htm

Figure 5.8 : Quelques images exemples dans la base de Fei-Fei

V.4.3 Tests

Pour valider notre système nous avons fait des tests sur plusieurs bases d'images telles que la base de **Z.Wang**, **Coil**, **Fei Fei li,** les résultats obtenus seront listés selon des catégories dans ce qui suit :

V.4.3.1 Temps d'indexation

Durant les tests appliqués, nous avons remarqué une grande différence entre les temps d'indexation des différents base selon la taille des images a traité et le nombre d'images à indexer.

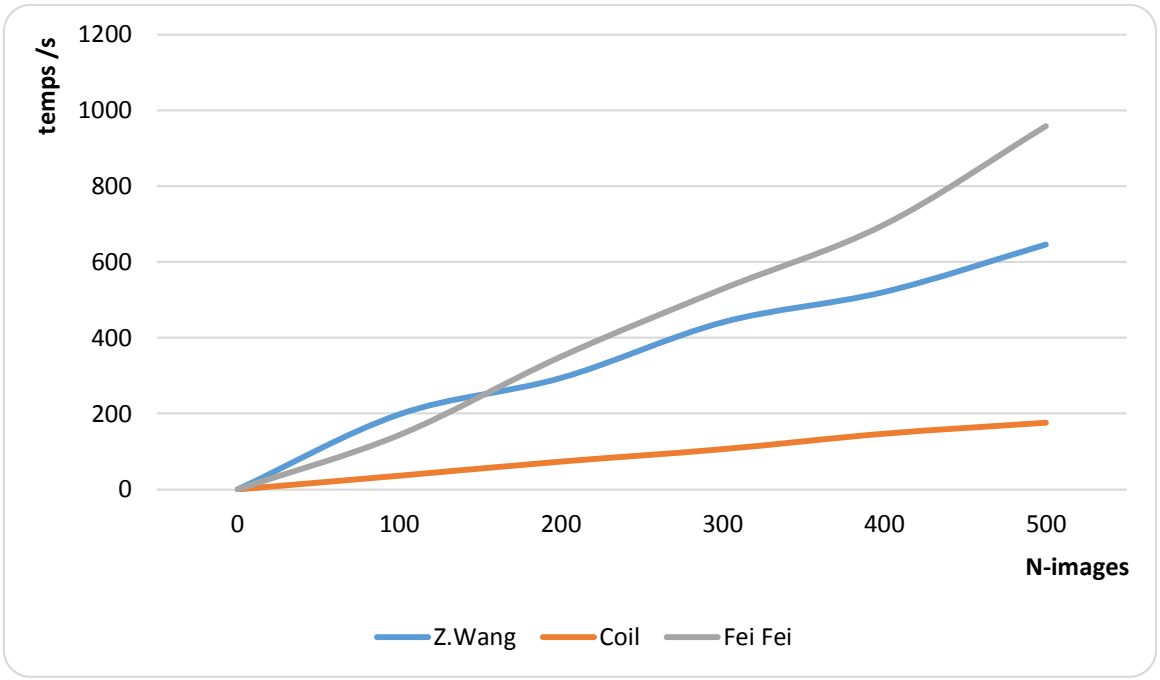


Figure 5.9 : temps d'indexation du descripteur SIFT

La base de **FeiFei** est la plus coûteuse en temps d'indexation avec le descripteur SIFT vu la taille et le type d'informations que les images de la base car on trouve que les images de cette base contient un plus grand nombre de descripteurs que les autre images d'autre bases.

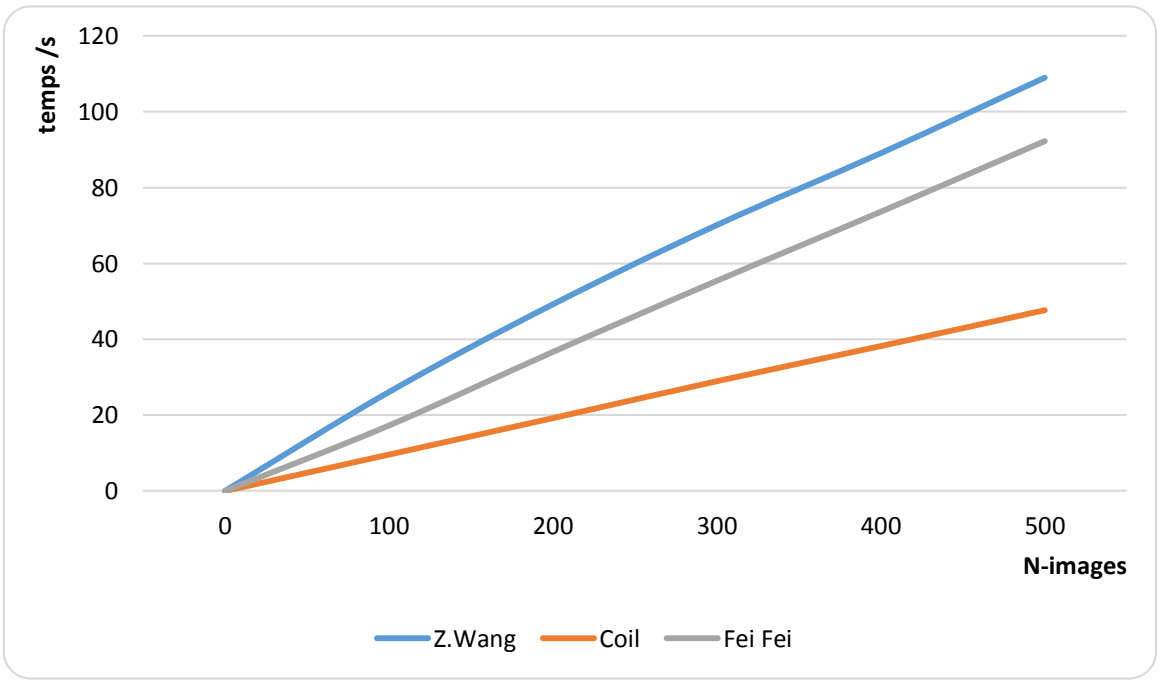


Figure 5.10 : temps d'indexation du descripteur LBP

La base de **Z.Wang** est la plus coûteuse en temps d'indexation avec le descripteur LBP car les images sont plus grandes au niveau de variété des textures qui contient ces images.

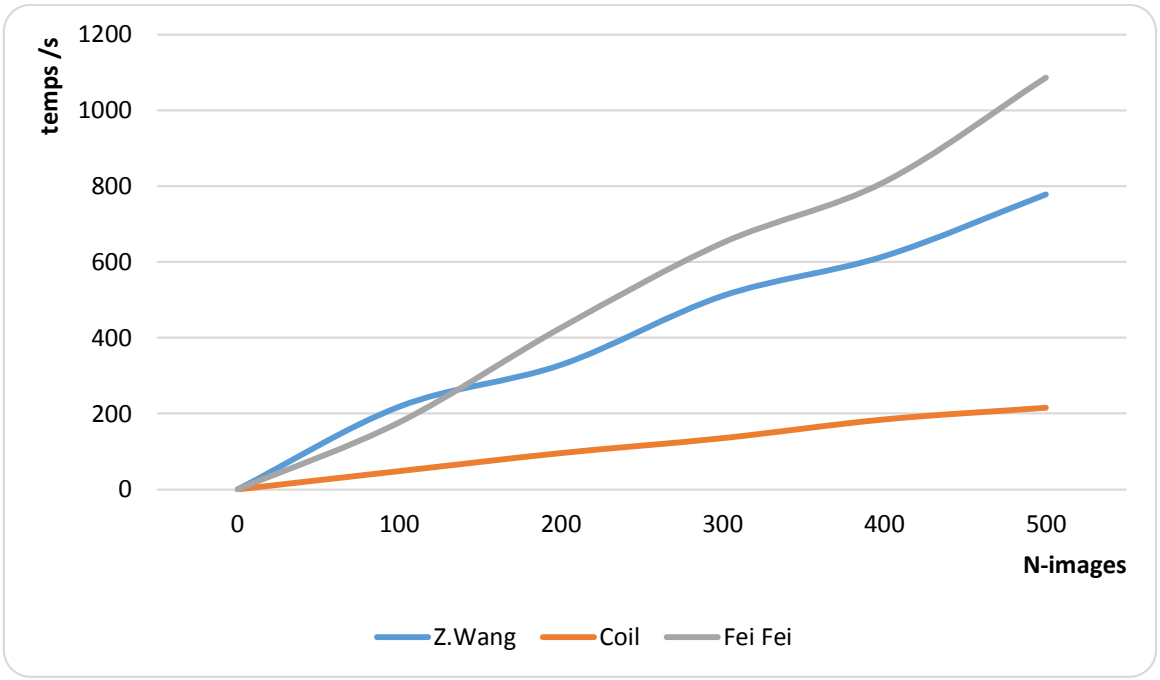


Figure 5.11 : temps d'indexation du descripteur SIFT+LBP

La base de **FeiFei** est la plus couteuse en temps d'indexation avec le descripteur SIFT+LBP a cause de nombre de descripteurs que ces images rend durant l'indexation et le type de ces information qui se varié beaucoup plus que d'autre bases d'images.

Dans tous les graphes précèdent on remarque que la base de **Coil** est la moins coûteuse en temps durant la phase d'indexation, tout ça est à cause de taille d'images et le type d'informations qu'elles contiennent.

V.4.3.2 Temps de recherche et Clustering

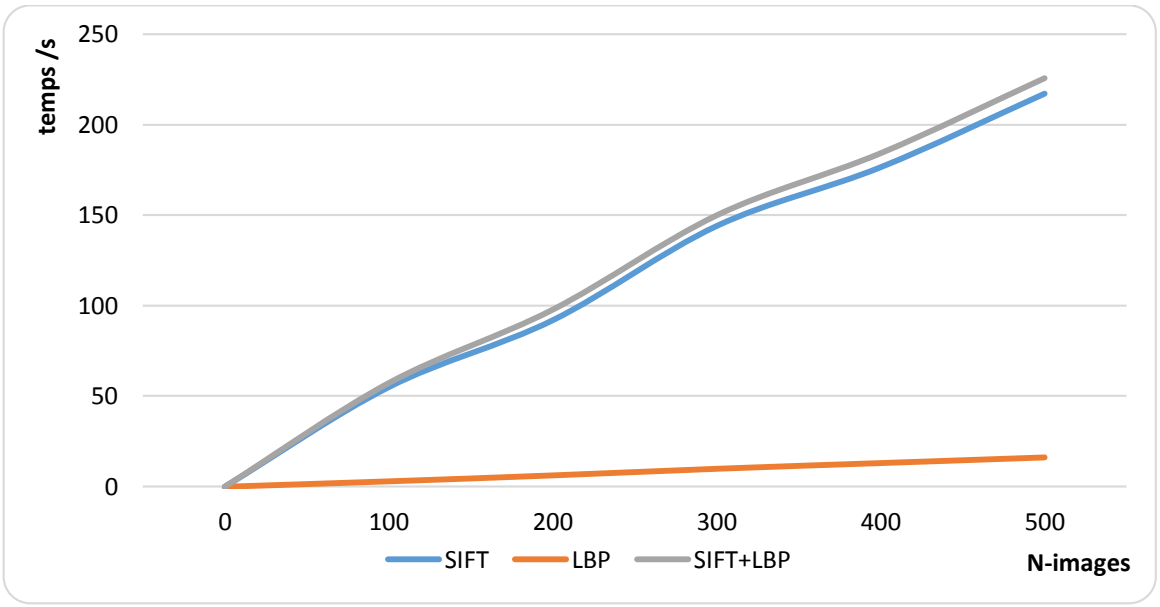


Figure 5.12 : temps de recherche dans utilisant la base de **Z.Wang** « clustering inclue »

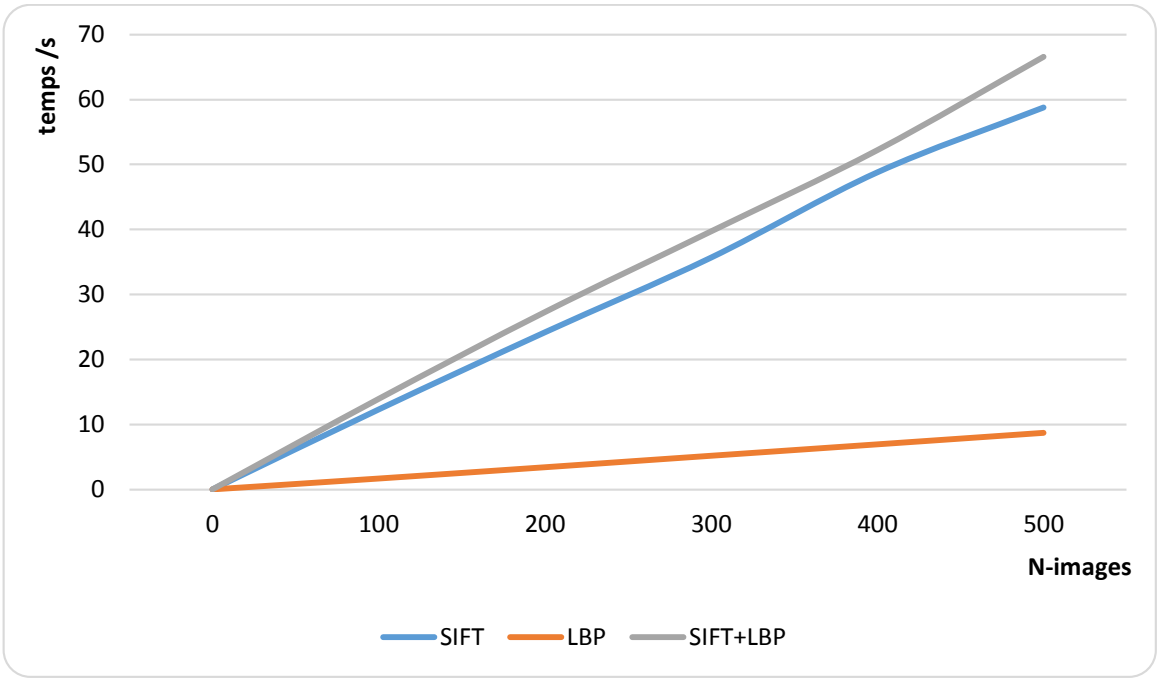


Figure 5.13 : temps de recherche dans utilisant la base de **Coil** « clustering inclue »

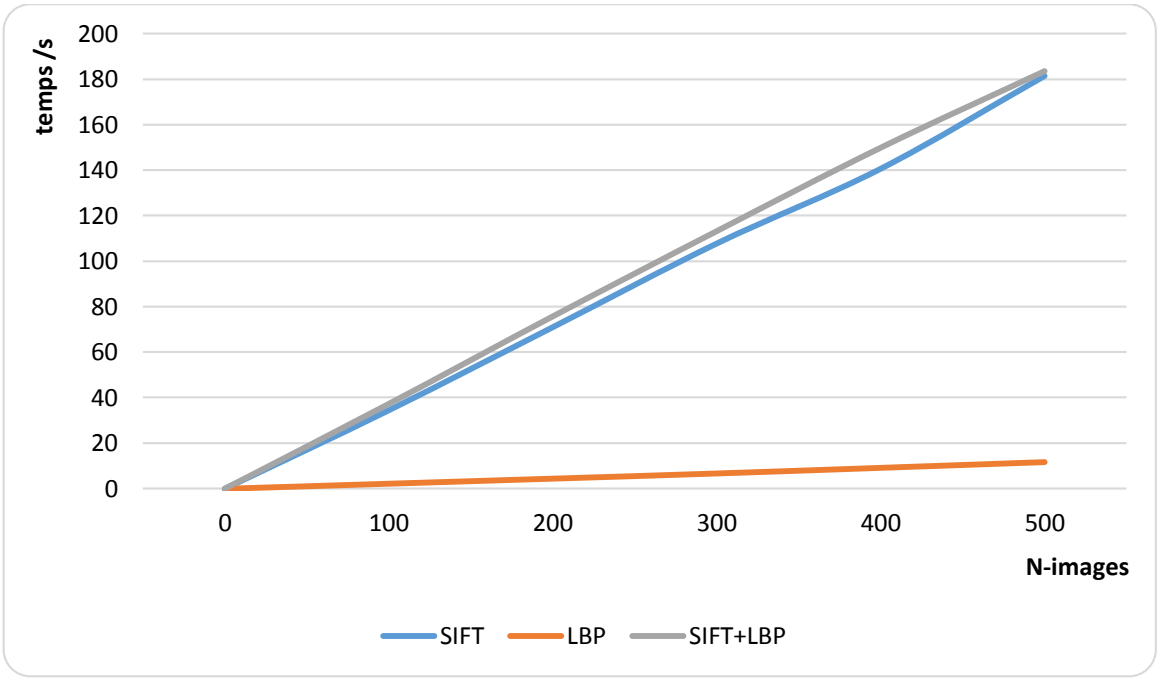


Figure 5.14 : temps de recherche dans utilisant la base de **Fei Fei** « clustering inclue »

La méthode de recherche la plus coûtante en temps est la méthode de SIFT+LBP, et la moins coûtante est avec LBP, aussi on a remarqué que le temps coûté en recherche avec les trois bases d'images utilisé est d'un ordre décroissant comme suit :

- La base de **Z.Wang.**
- La base de **Fei Fei.**
- La base de **Coil.**

La raison de ces résultats ont été venu de la masse d'information extraite de cette base avec la méthode SIFT où la variance des objets et couleurs joue un grand rôle dans l'augmentation de nombre de descripteurs de point d'intérêt retourné.

V.4.3.4 Pertinence du système face aux changements

Dans ce qui suit, nous avons présenté les différents résultats de pertinence du système face aux transformations géométriques sur les images soit en changement d'échelle et la rotation en utilisant des différents descripteurs.

V.4.3.4.1 Pertinence du système avec une indexation SIFT

Les tests sur les trois bases ont donné des bonne résultats avec la base de **Z.wang** et la base **Coil** et ont donné une bonne pertinence face aux changements, l'influence de contenu d'images a causé les résultats obtenu avec les requêtes de chaque base, aussi ces base d'images sont dédié a faire entrainer des systèmes d'intelligence artificiel « SIA » et que les images sont classé selon des type ou classes spécifique.

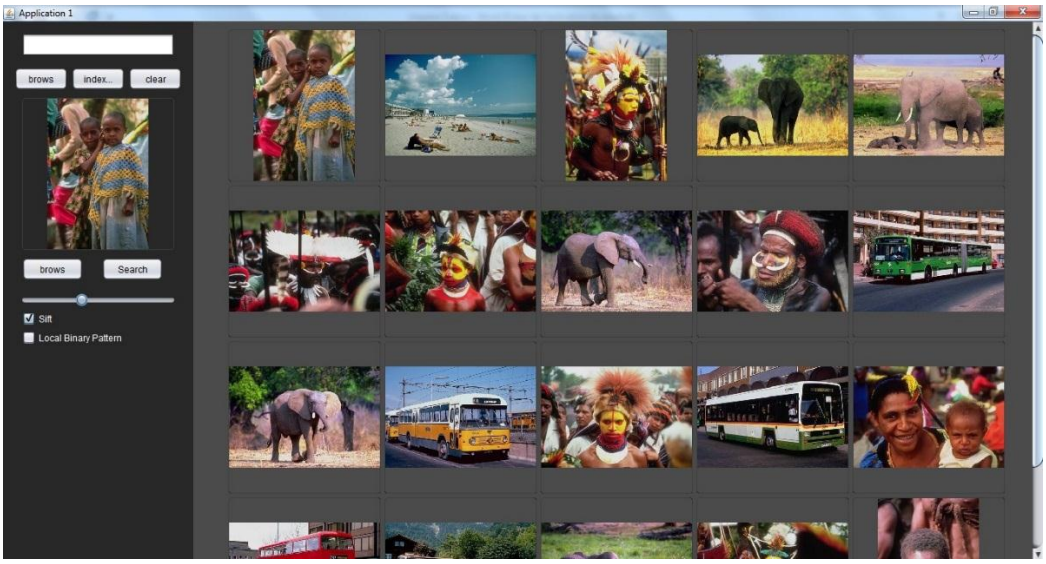


Figure 5.15 : résultats de pertinence du système avec méthode SIFT et la base de **Z.Wang**

V.4.3.4.2 Pertinence du système avec une indexation LBP

Les Bonnes résultats pour la méthode LBP ont été avec la base de **Coil**, les images de cette base sont les plus petits au niveau de taille et que les calculs arithmétiques sont minimisé à cause de l'invariance de type de motifs que contient ces images.

Figure 5.16 : résultats de pertinence du système avec méthode LBP et la base de **Coil**.

V.4.3.4.3 Pertinence du système avec une indexation SIFT-LBP

Les Bonnes résultats pour la combinaison des méthodes SIFT et LBP ont été avec la base de **Coil**, car cette base d'images est dédié a aidé dans les tests d'apprentissage automatique et vu le type d'images qui est le but de la création de la méthode SIFT qu'elle est robuste au attaques géométrique et les changements d'échelle.

Figure 5.17 : résultats de pertinence du système avec la combinaison des méthodes SIFT et LBP et la base de **Coil**.

V.3.4.3 résultats des tests de pertinence du système

Dans ce qui suit on va présenter les résultats pertinents des vingt premier images rendu par le système pour chaque image requête utilisé.

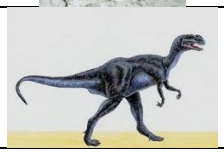
Image	*Méthode LBP*	*Méthode SIFT*	*Combinaison SIFT+Lbp*
Base Z.Wang			
	***3/20***	***9/20***	***6/20***
	***17/20***	***12/20***	***6/20***
	***5/20***	***10/20***	***12/20***
Coil			
	***4/20***	***20/20***	***17/20***
	***19/20***	***12/20***	***13/20***
	***13/20***	***7/20***	***5/20***
Fei Fei Li			
	***4/20***	***5/20***	***4/20***

Tableau 5.1 : les statistiques des différents résultats rendu par le système avec les bases d'images Z.Wang, Coil, Fei Fei Li

V.5 Conclusion

Durant ce chapitre, nous avons présenté les différentes fonctionnalités du système proposé pour l'indexation et la recherche d'images par le contenu, aussi les différents modules qui composent ce système.

Nous avons présenté aussi les résultats prisent durant les tests du système envers le temps écoulé pour l'indexation des différentes bases d'images utilisé pour les tests comme tel que **Z.Wang**, **Coil**, **FeiFei Li** et le temps de recherche écoulé pour accomplir une recherche avec les différentes méthodes tel que **SIFT**,**LBP** et la combinaison de **SIFT et LBP**.

A la fin de ce chapitre on a montré la pertinence du système face aux changements avec les trois méthodes d'indexation.

Conclusion Générale

Conclusion générale

L'indexation et la recherche d'images par le contenu sont des problèmes complexes et incontournables étant donnée la place que l'image numérique occupe à présent dans notre quotidien. Internet en est la meilleure illustration. Dans notre travail nous avons intéressés d'une part à l'indexation des images par le contenu et d'autre part à la recherche par le contenu.

Du point de vue de l'indexation, nous avons proposé un traitement automatique pour le calcul des indexes des images. Ce traitement se base sur l'analyse de la couleur et les transformations géométrique, où nous avons porté une attention à l'utilité des histogrammes LBP de couleurs et les descripteurs SIFT des points d'intérêt.

Du point de vue de recherche, nous avons utilisé le principe de classification des descripteurs où la recherche est par similarité entre ces descripteurs, et pour le calcul de cette dernière nous avons utilisé deux méthodes : le calcul de similarité par la corrélation.

Durant ce travail nous avons constaté deux points essentiels et cruciaux. Le premier, est que la couleur est une caractéristique discriminante d'une image, mais l'utilisation de la couleur tout seul dans un CBIR ne suffit pas, il faut rajouter autre descripteurs de texture et de forme pour améliorer les performances d'un CBIR.

Le domaine de recherche d'image par le contenu est un domaine très riche et exhibe une variance dans les techniques utilisées. Il n'existe pas une loi qui impose le choix d'une technique particulière pour l'extraction des informations depuis les images en se basant sur le contenu.

BIBLIOGRAPHIE

[1] **Jomier, G., Manouvrier, M., Oria, V. & Rukoz, M**. Indexation multi-niveau pour la recherche glo- bale et partielle d ’ images par le contenu 1. 1–20 (2002).

[2] **Omhover, J.-F. and Detyniecki, M**., Queries by visual content using fuzzy similarity Measures on regional descriptions. In: Proceedings of the European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems, pp. 43-44, 2004.

[03] **Hammache Arezki**, Recherche D ’ Information : Un Modèle de Langue Combinant Mots Simples et Mots Composés ,Thèse de Doctorat, Université Mouloud Mammeri de Tizi-Ouzou.

[04] **I. Atanassova**: Exploitation Informatique Des Annotations Sémantiques Automatiques d’Excom Pour La Recherche D'informations et La Navigation. Thèse de Doctorat Université PARIS-SORBONNEÉ. 2012.

[05] **M. A. Bourenane**. Un outil pour l’indexation des vidéos personnelles par le contenu. Thèse de Doctorat ,Université de Québec à trois- rivières., 2009.

[06] **G. Quellec**. Indexation et fusion multimodale pour la recherche d’information par le contenu. Application aux bases de données d’images médicales. Thèse de Doctorat, Université européenne Bretagne, Septembre 2008.

[07] https://fr.wikipedia.org/

[08] **A. Hafiane**. Caractérisation de textures et segmentation pour la recherche d’images par le contenu. Thèse de Doctorat ,Université de Paris-Sud XI, Décembre 2005.

[09] **O. Adjemout**, Reconnaissance automatique de formes à partir des paramètres morphologiques, de couleur et de texture : application au tri des graines de semences, thèse de magister, UMMTO, 2005.

[10] **J. Landre**: Analyse multi résolution pour la recherche et l’indexation d’images par le contenu dans les bases de données images – application À la base d’images paléontologique trans’tyfipal. Thèse de Doctorat, Université de Bourgogne U.F.R .. (2005).

[11] **M . Sonka**, HLAVAC V., BOYLE. R.: Image Processing, analysis and machine vision. pws publishing, seconde edition ed. 1999.

[12] http://signal2011.joora.fr/Tarizzo_Trang_imac3/page2.html

[13] Pearson, On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine. - 1901. - 6 : Vol. 2. - pp. 559–572.

[14] **Chandrasekaran. S, Manjunath. B. S, Wang. Y. F, Winkeler. J, and Zhang. H** An eigenspace update algorithm for image analysis: CVGIP: Graphical Models and Image Processing Journal, 1997.

[15] **Thierry Urruty** ; Optimisation de L'Indexation Multidimensionnelle : Application Aux Descripteurs Multimédia,Thése de doctorat, Universit´e des Sciences et Technologies de Lille, (2007).

[16] **B. Mohamed** ; modélisation statistique des collections d'image en utilisant les modèles de copules et application à l'indexation d'images. Université du Québec, (2011).

[17] **J. Fournier** ; indexation d'images par le contenu et recherche interactive dans les bases généralistes, Thèse de Doctorat, Université de Cergy-Pontoise, 2002.

[18] **David G. Lowe**, *Distinctive Image Features from Scale-Invariant Keypoints, 2004.*

[19] **J. E. Rodés**, " Détection et identification de points homologues par corrélation d'images", 2012.

[20] https://fr.wikipedia.org/wiki/Speeded_Up_Robust_Features; http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html.

[21] **Lotfi Houam** ; Contribution à L'analyse de textures de radiographies osseuses pour le diagnostic précoce de l'ostéoporose», Thèse de Doctorat, Université de Guelma.

[22] http://edutechwiki.unige.ch/fr/Clustering_et_classification_hi%C3%A9rarchique_en_text_mining.

[23] https://fr.wikipedia.org/wiki/Regroupement_hi%C3%A9rarchique.

[24] **H. Karima**, "Approche de partitionnement pour un apprentissage non supervisé des Usagers du Web (Amélioration de l ' approche k -means)," no. 1.

[25]. **Z. Guellil and L. Zaoui**, 'Proposition d'une solution au problème d'initialisation

cas du K-means', *CEUR Workshop Proceedings*, (2007).

[26] https://fr.wikipedia.org/wiki/Analyse_des_donn%C3%A9es.

[27] https://www.xlstat.com/fr/solutions/fonctionnalites/analyse-en-composantes-principales-acp

[27] http://wang.ist.psu.edu/.

[28] http://www1.cs.columbia.edu/CAVE/research/softlib/

[29] http://www.vision.caltech.edu/feifeili/Datasets.htm