



UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

EXPLORATION ET VISUALISATION DE L'ÉVOLUTION
TEMPORELLE D'UNE IMAGE DANS UN CORPUS VISUEL

MÉMOIRE
COMME EXIGENCE PARTIELLE
MAÎTRISE EN SCIENCES ET TECHNOLOGIES DE L'INFORMATION

PAR

AYMEN TALBI

NOVEMBRE 2016

Ce mémoire a été évalué par un jury composé des personnes suivantes :

Dr. Rokia Missaoui Présidente du jury

Dr. Nadia Baaziz Membre du jury

Dr. Moahand Said Allili Directeur de recherche

Dr. Nathalie Casemajor Codirectrice de recherche

le : 29 Novembre 2016

Remerciements

Je tiens tout d'abord à exprimer mes plus sincères remerciements à mon directeur de thèse, le prof. Mohand Said Allili, pour sa grande disponibilité, ses conseils, son soutien, sa patience infinie et son aide dans le cadre de ce travail de recherche et de rédaction. À la prof.

Nathalie Casemajor pour sa disponibilité, ses efforts et son encadrement précieux dans le cadre de ce travail.

Je tiens également à exprimer ma gratitude aux membres de mon comité de mémoire, les prof. Rokia Missaoui et Nadia Baaziz de l'Université du Québec en Outaouais pour leur participation à l'évaluation de ce travail.

Table des matières

Liste des abréviations, sigles et acronymes	iii
Résumé	iv
1 Introduction	1
1.1 Mise en contexte et problématique	1
1.2 Description du projet et ses objectifs	4
1.2.1 Corpus étudié	6
1.3 Organisation du mémoire	7
2 Revue de littérature	8
2.1 Analyse de corpus numériques visuels	8
2.1.1 Recherche d'images basées sur le contenu	9
2.1.2 Quelques outils du CBIR	12
2.2 Méthodes de visualisation du corpus visuel	16
2.2.1 Visualisation par graphes de dispersion 2D	16
2.2.2 Visualisation par graphes de relations	19
2.2.3 Visualisation par des grilles 2D/3D	22
2.2.4 Autres méthodes de visualisation	23
2.3 Limites de la visualisation du corpus visuel	25
2.4 Sommaire du chapitre	26
3 Caractéristiques de transformations d'images	27
3.1 Introduction	27
3.2 Transformations photométriques	28
3.2.1 Transformations du niveau de gris	28
3.2.2 Filtrage et réduction de bruit	32
3.2.3 Morphose (morphing)	34
3.2.4 Conversion du mode colorimétrique d'une image	35
3.2.5 Édition d'images	37

3.3	Transformations géométriques de l'image	39
3.3.1	Transformations affines	40
3.3.2	Transformations non-linéaires	41
3.4	Les métadonnées d'une image numérique	44
3.4.1	Définition et intérêts	44
3.4.2	Différents standards	45
3.4.3	Les outils de gestion des métadonnées d'images	48
3.4.4	Problèmes liés aux outils de gestion de métadonnées	48
3.5	Sommaire du chapitre	49
4	Méthode proposée et discussion des résultats	50
4.1	Introduction	50
4.2	Méthodologie proposée	51
4.2.1	Prétraitement des images du corpus	52
4.2.2	Identification des groupes de lignées d'images	53
4.2.2.1	Identification des relations fortes	54
4.2.2.2	Classification hiérarchique par distance entre histogrammes	58
4.2.2.3	Construction et visualisation des relations de précedence	64
4.2.3	Analyse des éditions d'images	65
4.2.3.1	Détection de la présence d'édition	68
4.2.3.2	Catégorisation probabiliste des éditions	74
4.2.4	Intégration de métadonnées dans la classification	79
4.3	Sommaire du chapitre	81
5	Conclusion générale	82

Liste des abréviations, sigles et acronymes

URL Uniform Resource Locator

RVB Rouge, Vert, Bleu

CM Cyan, Magenta, Yellow

TSL Teinte, saturation, luminosité

EXIF Exchangeable image file format

IPTC International Press Telecommunications Council

XMP Extensible Metadata Platform

JEIDA Japan Electronic Industries Development Association

GPS Global Position System

RDF Resource Description Framework

XML Extensible Markup Language

CBIR Content Based Image Retrieval

CBCR Content Based Copy Retrieval

CBPIR Content Based Painting Image Retrieval

SIB Semantic Image Browser

DOM Document Object Model

Résumé

Les images circulant sur les médias sociaux sont produites et partagées entre les sites Web de manière rapide et exponentielle. Cependant, l'énorme quantité d'images existantes engendre des problèmes quant à leur organisation et au traçage de leur évolution sur le Web. Les travaux de recherche dans ce domaine ont été pour la plupart proposés pour traiter des grandes collections d'images à des fins de recherche et/ou de visualisation d'images. Ces derniers s'avèrent peu efficaces pour grouper et ordonner des copies d'une même image déclinée en différentes copies sur le Web. En effet, dans plusieurs domaines, tels que l'analyse des phénomènes artistiques, politiques, ethniques, culturels, scientifiques ou autres, il est important de comprendre les éléments visuels partagés par les utilisateurs du Web et de suivre leurs évolutions dans l'espace et le temps.

L'objectif principal de ce travail est de développer une méthodologie permettant de classer automatiquement les copies et variantes de la même image pour déterminer la séquence chronologique de leur création. En d'autres termes, elle permettra de distinguer les différentes relations temporelles et spatiales qui relient une image originale à ses reproductions. Ces relations et variations peuvent être organisées visuellement sous forme de topographies arborescentes illustrant le réseau de dissémination sur le Web de chaque image depuis sa création. Nous proposons une méthode basée sur l'analyse des caractéristiques géométriques et photométriques de l'image pour estimer le meilleur ordonnancement temporel des images. Des expérimentations sur un ensemble d'images représentant des copies d'une photographie de *Gustave Doré* ont montré la validité de notre approche.

Chapitre 1

Introduction

1.1 Mise en contexte et problématique

Au cours des dernières décennies, la croissance du Web a créé une nouvelle situation où des centaines de millions de personnes et d'internautes créent et partagent régulièrement du contenu visuel (photos, vidéos, etc.). Selon les dernières statistiques [25], plus de 150 milliards pages Web ont été créées durant la période de 1996 jusqu'à aujourd'hui, et des centaines de milliards d'heures de vidéos ont été postées sur YouTube. Selon les statistiques fournies par Facebook au début de l'année 2012, 7 milliards d'images sont téléversées par les utilisateurs de ce réseau social chaque mois.

Grâce à la disponibilité des caméras numériques à bas prix et au développement rapide des technologies de l'information, l'utilisation à grande échelle des images numériques a commencé à se répandre sur le Web et les médias sociaux comme une nouvelle forme de communication socio-culturelle [25, 34]. Les images circulant sur ces médias sociaux sont produites et partagées entre les sites Web de manière rapide et exponentielle. Cependant, l'énorme quantité d'images existantes engendre des problèmes quant à leur organisation et au traçage de leur évolution sur le Web. Ceci, à son tour engendre des difficultés pour rechercher des images pertinentes pour les

utilisateurs ainsi que pour comprendre l'organisation spatio-temporelle de groupes d'images interreliées. Plus généralement, le problème de la gestion des grandes collections d'images se pose dans plusieurs domaines, tels que la recherche d'images basée sur le contenu [21, 60, 24, 56], le groupement et la classification d'images [33] et la reconnaissance d'objets et d'événements dans les images (personnes, visages, etc.) [9, 43, 44]. La gestion des collections peut aussi servir à suivre les copies d'images à des fins de sécurité (par exemple, pour la protection des droits d'auteur) [67] ou simplement à des fins de visualisation (notamment sous forme de graphes retraçant les tendances artistiques, la distribution géographique du contenu visuel, etc.) [62, 39].

Récemment, avec la prolifération des réseaux sociaux, du commerce électronique et de l'utilisation d'Internet pour la communication à grande échelle, le besoin de suivre la trace des contenus visuels à des fins de sécurité ou d'analyse socio-culturelle devient très important [10, 11, 38]. Pour la sécurité, l'apparition de la biométrie comme moyen d'identification d'individus a engendré des systèmes pour la reconnaissance faciale sur les images et les vidéos [26]. D'ailleurs, les méthodes récentes dans ce domaine sont de plus en plus capables de prendre en charge la variation des aspects visuels de personnes tels que le vieillissement, la couleur de la peau, le sexe, etc [2]. La sécurité concerne aussi le suivi d'images sur le Web pour la protection des droits des créations visuelles (commerciales et artistiques). Des travaux ont été proposés, par exemple, pour détecter des doublons (proches) ou des versions forgées ou dérivées d'images originales [11, 33]. Ces travaux proposent des méthodes permettant d'analyser les images et de détecter les transformations photométriques et géométriques qui y sont opérées. La Figure (1.1) montre un exemple d'une image originale (a) et sa version (b) ayant subi une *rotation*, un *changement d'échelle* et un *cisaillement*.

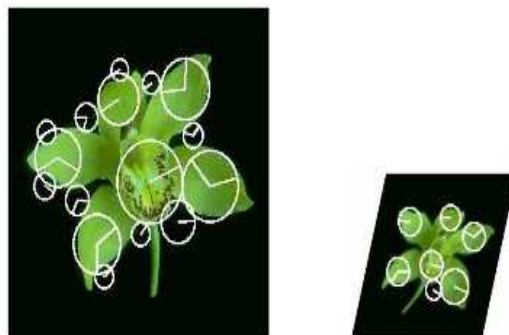


FIGURE 1.1 – Exemple de détection de *copyright* [33]

Avec la prolifération des médias sociaux, l'analyse des collections d'images est aussi devenue un outil indispensable pour comprendre les échanges de données sur les réseaux sociaux et la communication sur le Web [36, 38]. Par exemple, l'analyse temporelle de productions artistiques visuelles peut renseigner sur l'évolution des tendances et des styles de création à travers les années et les pays [7, 39]. Elle permet aussi d'analyser les relations entre individus partageant les mêmes intérêts et, ainsi, inférer des informations de proximité et de groupes entre individus [24].

La plupart des travaux de recherche récents dans ce domaine visent à créer des outils de visualisation pour comparer de grandes collections d'images, en général, et d'œuvres artistiques en particulier [36, 38, 39, 62]. Dans son travail, Manovich [39] a développé un outil utilisant les statistiques d'images pour visualiser et comparer des peintures d'artistes sur le Web. D'autres travaux ont proposé des visualisations sous forme de graphes afin de comprendre les relations entre les images d'une collection [62, 66, 61]. Zhang et al. [66] ont étudié l'intégration des transitions animées dans la visualisation des collections d'images. Plus récemment, Wang et al. [30] ont développé un outil appelé *IGraph* qui intègre plusieurs étapes pour suivre les relations entre des images et leurs annotations textuelles. Dans d'autres travaux, les auteurs Camargo et al. [20] et Yang et al. [64] ont utilisé l'analyse sémantique pour visualiser les collections d'images et ainsi permettre la recherche d'images en fonction de leur

contenu sémantique.

Les travaux mentionnés ci-dessus ont été pour la plupart proposés pour traiter des grandes collections d'images à des fins de classification et/ou de visualisation d'images. Bien qu'ils soient efficaces pour classer des images en groupes homogènes à partir de leurs caractéristiques visuelles (par couleurs, textures, etc.), ils s'avèrent peu efficaces, voire incapables, de grouper et d'ordonner des copies d'une même image déclinée en différentes versions sur le Web. En effet, dans plusieurs domaines, tels que l'analyse des tendances de groupes artistiques, politiques, ethniques, culturels, scientifiques ou autres, il est important de comprendre les éléments visuels partagés par les groupes sociaux et de suivre leur évolution dans l'espace et le temps. Ceci est tout aussi utile pour déterminer le rôle de différents acteurs dans les groupes ainsi que pour cerner les conditions de diffusion et d'appropriation de nouvelles créations, notamment artistiques (par exemple, des peintures et photographies). La croissance rapide de l'utilisation du Web et la disponibilité des données rendent possible le développement de techniques d'analyse automatique et à grande échelle d'images pour apporter des solutions plus adaptées à ce genre de problématique.

1.2 Description du projet et ses objectifs

Le but de ce mémoire est de développer une méthode automatique permettant de classer les copies et variantes de la même image pour déterminer la séquence chronologique de leur création. En d'autres termes, cette méthode permettra de distinguer les différentes relations temporelles et spatiales qui relient une image originale à ses reproductions. Ces relations et variations peuvent être organisées visuellement sous la forme de topographies (ou graphe de précédence) illustrant le réseau de dissémination de chaque image. Afin d'atteindre cet objectif du projet, nous avons défini une nouvelle approche qui s'échelonne en deux phases.

1. **Analyse du contenu visuel des images** : consiste à utiliser des techniques de traitement d'images numériques afin de détecter les différentes variations et copies de l'image originale dans notre corpus. Ceci permettra la détection des différentes transformations géométriques et photométriques appliquées aux images originales.
2. **Analyse des métadonnées** : consiste à analyser les données numériques et textuelles utiles (ex. les dates de création et de modification du fichier) permettant de suivre l'évolution des copies d'images. Une fois générées à partir de notre corpus, ces métadonnées seront traitées si possible pour extraire les relations temporelles entre les différentes variantes.

Le sujet de ce mémoire s'inscrit dans un cadre plus large d'un projet de recherche intitulé : *Trajectoires d'images en réseau : logiques d'usage et traces matérielles sur le Web*, supervisé par la professeure Dr. Nathalie Casemajor en collaboration avec le professeur Dr. Mohand Saïd Allili. Ce projet s'intéresse aux trajectoires d'un échantillon d'images numériques en circulation sur le réseau Internet. Il étudie notamment la dimension matérielle des traces d'usages sur le Web et les réseaux de diffusion spatio-temporelle des contenus.

Un des volets de ce projet qui ne sera pas abordé dans ce mémoire est l'analyse du paratexte des différentes pages Web contenant les images du corpus, en complément à l'analyse visuelle et les métadonnées des images. Ce volet s'appuie sur une analyse qualitative du contenu des pages Web, visant notamment à contextualiser les images. À partir des URLs, chaque page où l'image originale a été republiée sera identifiée selon une catégorisation typologique (institution muséale, presse, blogue, etc.), une catégorisation thématique des sites (spécialisés en art visuel ou généralistes) et un repérage des dates de publication (si disponibles).



FIGURE 1.2 – Quelques copies de corpus étudié

1.2.1 Corpus étudié

Le corpus sur lequel porte notre étude contient 60 images qui sont toutes des variantes d'une photographie ancienne. Il s'agit d'un portrait du graveur français Gustave Doré, pris par le photographe Nadar entre 1856 et 1858, dont le négatif original est conservé à la Médiathèque de l'architecture et du patrimoine (France). L'ensemble des images constituant ce corpus ont été collectées entre 2010 et 2015. Un premier ensemble a été collecté via une recherche dans les principales bases de données patrimoniales françaises (notamment la Médiathèque de l'architecture et du patrimoine et la Réunion des musées nationaux) et dans Wikimedia Commons. Un autre ensemble a été collecté grâce à une recherche indexée à travers le moteur de recherche Google Images avec les mots-clés '*Gustave Doré Nadar*'. La Figure (1.2) montre quelques exemples de cet échantillon.

Le corpus obtenu est constitué des multiples variantes avec différentes caractéristiques, tel que le format des images, les transformations photométriques appliquées sur chacune (changement d'espace de couleur, luminosité, éditions d'images soit par insertion de texte ou de cadre, etc.).

1.3 Organisation du mémoire

Le reste de ce mémoire est organisé comme suit : le deuxième chapitre présente une revue de littérature à propos des différentes techniques de constitution et de délimitation des corpus visuels, ainsi que les diverses approches de visualisation de ces derniers. Dans le chapitre 3, nous allons présenter les différentes transformations photométriques, géométriques ainsi que les différents standards de métadonnées d'une image numérique. Le chapitre 4 décrit la méthode proposée et les résultats obtenus. Enfin, le chapitre 5 conclut le mémoire en mentionnant les limites de notre approche et en proposant quelques pistes pour améliorer notre solution.

Chapitre 2

Revue de littérature

Introduction

Le problème de classification, de suivi et de visualisation des contenus visuels sur le Web a fait l'objet d'une recherche considérable depuis les dernières années [18, 30, 40, 62]. Ce chapitre présente une brève revue de la littérature sur les différentes méthodes de recherche et de visualisation des collections d'images numériques. Comme il existe de nombreux travaux dans ces domaines, nous allons retenir les plus récents et ayant eu le plus d'impact dans la communauté scientifique. Cette revue de littérature est divisée en deux parties : la première partie décrira les différents travaux traitant du problème de constitution et délimitation des corpus et du suivi des contenus visuels sur le Web (c'est-à-dire la recherche d'images basée sur le contenu), la deuxième partie résumera les différentes méthodes de visualisation de corpus visuels.

2.1 Analyse de corpus numériques visuels

Hormis la recherche et la localisation de documents visuels sur le Web, il est important d'analyser des groupes (corpus) d'images afin d'extraire différentes informations utiles (sous groupes, ordre chronologique, provenance, etc.). Parmi les

champs d'analyse du Web, la recherche sur les cultures numériques visuelles prend une place importante avec le développement des réseaux sociaux et l'expansion du Web comme moyen de communication rapide et largement accessible [24, 25]. Celle-ci a pour principal objectif de cartographier le contenu de corpus visuels sur le Web et de comprendre leurs usages sociaux, en s'appuyant sur leur évolution dans l'espace et le temps. Cependant, le très grand nombre d'images existant sur le Web et la diversité de leur contenu visuel rend cette tâche très difficile à réaliser manuellement.

Pour apporter une solution à ce problème, de plus en plus d'approches basées sur l'analyse automatique des images ont été proposées [30, 38, 39]. Les récents développements dans l'imagerie et l'apprentissage par ordinateur ont apporté des améliorations importantes dans la recherche et la classification d'images [21, 50, 3, 56] ainsi que la reconnaissance de leur contenu visuel [5, 6, 4]. Il est à noter cependant que, quelque soit le problème d'analyse visé, les chercheurs sont d'abord confrontés au problème de la délimitation spatiale et temporelle des frontières du corpus visuel sur lequel l'analyse sera basée. Autrement dit, il est nécessaire de cerner sur le Web le(s) groupe(s) d'images pertinentes sur lesquelles l'analyse portera. Ce problème de recherche est communément nommé *la recherche d'images basée sur le contenu* (CBIR) [56]. L'objectif étant de trouver des méthodes pour rechercher et identifier les images pertinentes à une requête constituée d'un ou plusieurs exemples d'images et/ou de leurs annotations.

2.1.1 Recherche d'images basées sur le contenu

De nos jours, les images jouent un rôle très important dans plusieurs domaines (médical, météorologie, journalisme, publicité, arts, médias sociaux, etc.). Cependant, l'existence d'un très grand nombre d'images rend difficile la constitution manuelle des métadonnées précises grâce auxquelles elles peuvent être indexées et recherchées de manière efficace [56]. Le domaine de la recherche d'images par le contenu

a vu le jour dans les années 1980, et il s'est accéléré avec l'avènement du Web, pour concevoir des systèmes de recherche et d'indexation d'images basés sur le contenu visuel de ces dernières. Par exemple, ces systèmes peuvent être requis pour chercher à partir d'une requête des images ayant le même contenu sémantique (objets, scènes, etc.) [9, 64] ou des copies d'une image pour analyser les modifications qui y sont apportées [11, 67]. Les systèmes actuels de recherche d'images pertinentes à la constitution d'un corpus d'analyse peuvent être classés en deux catégories principales :

1. La recherche à base d'une requête textuelle.
2. Les systèmes basés sur le contenu, qui utilisent la similarité visuelle entre images.

Il existe aussi des approches hybrides qui combinent les deux types de systèmes. Ces dernières permettent d'associer des mots-clés aux caractéristiques de bas niveau d'images (couleur, texture, etc.), tels que l'approche CHABOT [42] qui permet une association d'un mot-clé à un ensemble de caractéristiques de couleur. Comme le montre la Figure (2.1), généralement le processus de recherche d'images commence par une requête de l'utilisateur qui est composée d'exemples d'images et/ou des annotations. Quelques techniques de prétraitement peuvent être appliquées (segmentation, détection de points d'intérêts, redimensionnement, filtrage, etc.) dépendamment des caractéristiques recherchées des images [56]. Les images de la base de données sont alors triées par ordre de similarité de leurs caractéristiques avec la requête, et le système retournera les images les plus similaires considérées pertinentes au besoin de l'utilisateur.

Pour la recherche d'images à base de requêtes textuelles, il s'agit d'un service disponible dans la majorité des moteurs de recherche actuels (Google, Yahoo, etc.). Lorsqu'un utilisateur lance une requête textuelle, les images indexées par les mots-

clés composant la requête sont prises en considération au cours de la recherche [7]. Cependant, ces méthodes dépendent grandement des annotations associées aux images. En effet, à cause d'annotations non précises ou de problèmes de synonymie et de polysémie, ces moteurs peuvent renvoyer des résultats inappropriés ou non précis pour la requête textuelle. Par ailleurs, du fait que l'annotation est faite par l'intervention humaine, cela rend l'indexation d'une même image différente à cause de la subjectivité humaine. En résumé, le problème majeur de la recherche basée sur les annotations reste la difficulté d'assurer une annotation précise pour un très grand nombre d'images [50, 56].

Pour surmonter la limitation de l'annotation textuelle, les chercheurs ont proposé une deuxième approche qui consiste à analyser visuellement le contenu des images pour en extraire des représentations fidèles et précises de leur contenu sémantique [7, 64]. La recherche d'images basée sur le contenu (CBIR) a l'avantage de pallier au manque de précision de l'annotation textuelle, en utilisant directement le contenu de l'image pour la recherche et l'indexation [56]. Depuis les premières méthodes proposées au début des années 1990 [56], la CBIR a fait l'objet de plusieurs recherches jusqu'à ce jour [9]. Plusieurs systèmes ont été proposés, tels que : CQIB [41], Visual-Seek [57], Simplicity [60], et Blobworld [21]. Ces systèmes se basent sur l'extraction des caractéristiques de bas niveau de l'image (couleur, texture, etc.) afin de représenter le contenu visuel des images. La limite des méthodes de CBIR de première génération est qu'elles utilisent des caractéristiques d'images de bas niveau, qui sont pour la plupart dépourvues de sémantique visuelle. En effet, deux images ayant des contenus visuels similaires, par exemple leur couleur, peuvent appartenir à des contextes sémantiques différents. Cette limite est communément appelée : *problème du vide sémantique (semantic gap)* en RIBC [50].

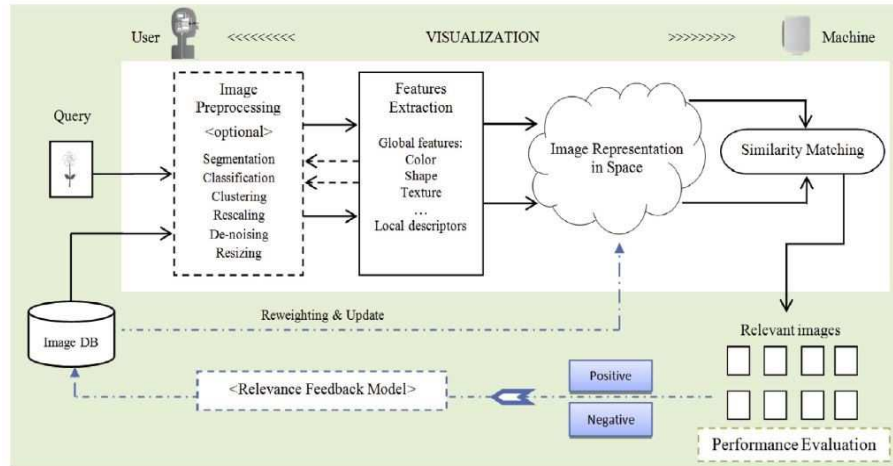


FIGURE 2.1 – Cadre général du système CBIR [56]

Les récents développements de la reconnaissance visuelle ont conduit à de nouvelles méthodes d'analyse d'images de plus en plus riches en termes de contenu sémantique (type d'objets, scènes, annotations, etc.) [50]. Pour la reconnaissance d'objets, par exemple, des travaux récents proposent des techniques basées sur les réseaux de neurones pour apprendre sur des millions d'images la détection et la reconnaissance d'objets dans des images couleurs [4]. D'autres méthodes ont aussi été proposées pour la reconnaissance des types de scènes présents sur une image (scène urbaine, paysage forestier, mer, ciel, etc.) [6, 4]. Ces techniques, avec l'annotation manuelle, permettent une représentation plus précise du contenu sémantique visuel, rendant ainsi la recherche d'images plus précise et efficace pour le besoin des utilisateurs [3].

2.1.2 Quelques outils du CBIR

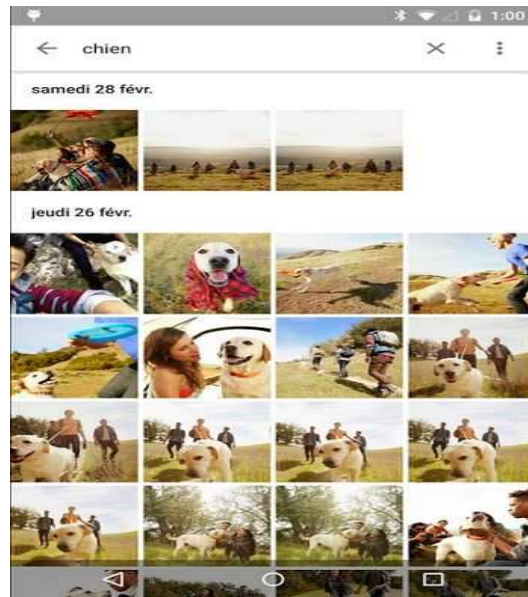
Plusieurs systèmes CBIR sont développés comme des prototypes de recherche. Certains ont été rendus disponibles pour le public comme option alternative de recherche d'images. Parmi ces systèmes, on peut trouver *TineEye* et *Google Images* [9]. La littérature classe ces deux exemples comme des leaders dans le domaine. Ces outils offrent aux utilisateurs plusieurs techniques pour soumettre l'image re-



FIGURE 2.2 – Technique de soumission d’une image requête avec *Google Images*

quête, tels que le téléchargement, l’utilisation de liens Web localisant l’image ou des mécanismes *de drag and drop* des images. La Figure (2.2) présente un exemple de ces techniques : on peut voir que *Google Images* utilise la possibilité de saisir l’adresse URL de l’image ou de l’importer. *Google Images* et *TinEye* compilent une liste d’images ou de figures même fortement modifiées (déformées, recadrées, compressées, redimensionnées, etc.) ressemblant à l’image recherchée. *Google Images* a la possibilité de trouver plus d’images car il indexe plus de sites que *TinEye*, mais l’avantage de ce dernier est qu’il a la capacité de retrouver des images plus fortement modifiées [24].

En 2015, Google a développé une application de gestion de photos intitulée *Google Photos*. Grâce aux techniques d’apprentissage automatique, cet outil permet de regrouper les images d’une manière automatique selon plusieurs techniques de recherche, notamment en les classant par catégories de personnes, de lieux, et d’objets présents sur les photos. Par défaut, *Google Photos* range les photos par ordre chronologique, de la plus récente jusqu’à la plus ancienne. L’une des techniques de recherche et de regroupement des images offerte par cet outil est la recherche par personne, grâce à la création de modèles de visages dans l’ensemble des photos, afin de regrouper les visages similaires. L’utilisateur peut aussi créer une étiquette pour

FIGURE 2.3 – Recherche avec *Google Photos*

chaque personne apparaissant dans les photos, en lui attribuant un nom ou un identifiant. Une fois les visages identifiés, l'utilisateur peut effectuer une recherche avec le nom d'étiquette choisi. Une autre technique de recherche des images est centrée sur les lieux, soit en utilisant les données de géolocalisation, soit par la saisie manuelle du lieu de la prise de vue des photos. De plus, on peut effectuer une recherche par thème (voiture, plage, selfie, etc.). La Figure (2.4) montre un exemple de détection d'ensemble d'images par la recherche avec le mot *chien*.

Récemment, les chercheurs se sont penchés sur le problème de récupération de copie basée sur le contenu (CBCR), qui vise à récupérer toutes les versions modifiées ou les versions précédentes d'un objet candidat donné (image, vidéo, etc.). Les méthodes de recherche basées sur le contenu dédiées à la détection des copies sont apparues ces dernières années dans plusieurs domaines et applications tels que la surveillance et la protection du droit d'auteur [13], [14], [15]. Bien que le problème de détection de copie d'image puisse être aussi considéré comme un aspect de la récupération d'image basée sur le contenu (CBIR), il existe une différence fonda-

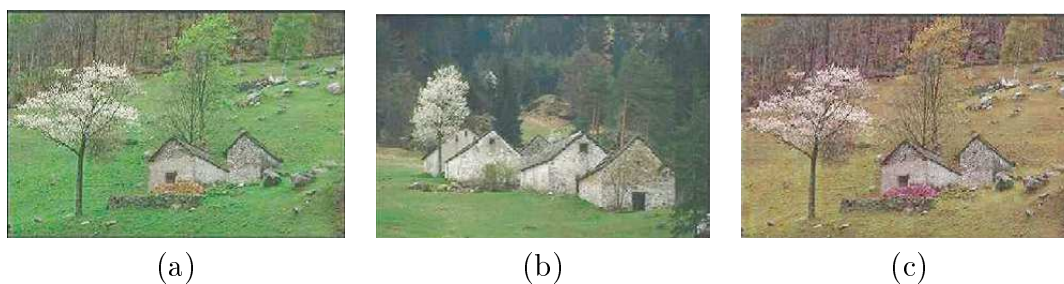


FIGURE 2.4 – Exemple de différence entre la récupération par couleur et la détection de copie basée sur le contenu : (a) une image originale, (b) une image similaire à l'image originale en termes de couleur, (c) une image qui a subi un changement de teinte

mentale entre la récupération d'image basée sur le contenu et la détection de copie d'image. Le détecteur de copie d'image recherche toutes les copies d'une image de requête, alors qu'un système de recherche d'images basé sur le contenu recherche des images similaires, généralement en termes de couleur [16]. Par exemple, La figure 1 montre un exemple de différence entre les systèmes CBIR et CBCR, dont (a) est une image originale (ou requête) et (b) est une image similaire à la plus à gauche en termes de couleur, l'image (c) est une image qui a subi un changement de teinte par rapport à l'image originale [16]. Pour l'image de requête en (a), un système de récupération basé sur le contenu (CBIR) récupérerait l'image en (b) d'abord, alors qu'un détecteur de copie (CBCR) devrait reconnaître l'image en (c) comme une copie de l'original.

Les outils présentés précédemment (Google Images, TinEye et Google Photos) sont très performants dans la constitution de corpus d'images similaires à une image source. Sur la base de l'analyse de l'image de départ, ces outils permettent de détecter même les images modifiées et retravaillées à l'aide de logiciels de retouche [24]. Après la présentation de différents approches et outils de constitution et délimitation des corpus d'images, nous allons présenter dans la section suivante les différentes méthodes de visualisation de ces corpus visuels.

2.2 Méthodes de visualisation du corpus visuel

Avant l'avènement du numérique, les livres, les musées et les médias organisaient visuellement leurs documents et données selon un petit nombre de catégories distinctes (périodes, écoles artistiques, nom d'auteur, etc.) [37]. De nos jours, la classification manuelle devient incapable de traiter les grands volumes de contenus numériques visuels. Depuis les années 1960, les chercheurs ont mis au point des techniques pour décrire automatiquement les propriétés visuelles des images. Ceci a permis de développer des méthodes pour l'organisation et la visualisation des collections d'images en diverses formes graphiques [62]. Par ailleurs, le besoin d'explorer et de visualiser les ensembles d'images devient de plus en plus pressant afin de comprendre les relations entre les images, ainsi que pour analyser l'évolution spatiale et temporelle de ces dernières [48]. La partie suivante décrit quelques techniques existantes dans la littérature pour la visualisation de corpus visuels.

2.2.1 Visualisation par graphes de dispersion 2D

L'une des méthodes utilisées pour la visualisation des collections d'images est la visualisation en graphes de dispersion 2D (planaires), où chaque axe présente une caractéristique visuelle (couleur versus entropie, luminosité versus écart type, etc.). Swain et al. [58] ont utilisé la couleur afin de visualiser les images les plus similaires. Gagaudakis et al. [28] ont utilisé la forme de texture pour visualiser une collection d'images selon la valeur des descripteurs utilisés. Manovich et al. [37] ont développé une technique de visualisation de styles de peintures d'un artiste. L'approche est inspirée de logiciels commerciaux tels que Google Web Analytics et Google Trends [37]. Les peintures sont visualisées selon leurs caractéristiques visuelles (luminance, saturation, etc.) par rapport à l'axe du temps. Le but étant de distinguer les catégories de styles et les périodes composant la carrière d'un artiste. La Figure (2.5) montre un exemple de visualisation de peintures tiré du travail de Manovich. L'axe

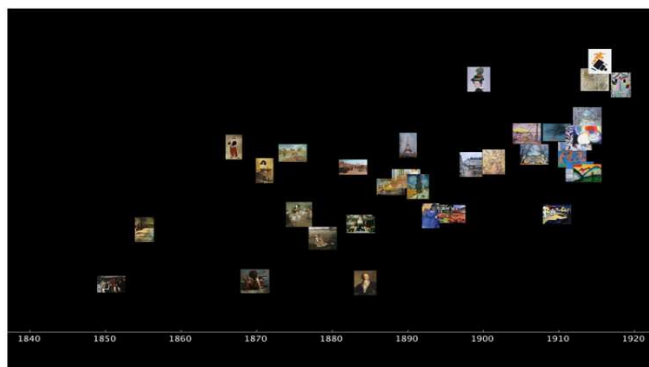


FIGURE 2.5 – Visualisation des oeuvres de 35 peintures [37]

des abscisses et l'axe des ordonnées représentent les dates des tableaux et la valeur inversée de leur asymétrie. Une peinture ayant des tons clairs aura une valeur d'inclinaison négative, tandis qu'une peinture avec des tons sombres aura une valeur d'inclinaison positive.

Dans un second travail, Manovich [37] a proposé une approche pour surmonter les limites des interfaces Web populaires (Flickr, Photobucket, Google Docs, Excel, etc.) qui ne sont pas capables de visualiser une grande collection d'un seul coup, en fonction de leurs caractéristiques visuelles ou sémantiques. Pour ce faire, l'approche proposée, appelée *media visualisation (visualisation de médias)*, combine les fonctionnalités des applications de visualisation des médias afin de rendre visible la forme ou la silhouette générale de la collection d'images. Contrairement aux autres outils de visualisation de données qui transforment en premier lieu les contenus en nombres pour pouvoir les visualiser, cette méthode permet de créer directement de nouvelles représentations à partir des données visuelles des collections.

Dans un autre travail, Manovich [40] a utilisé le concept *d'espace de style* pour comparer différents groupes d'images de peinture et suivre leur évolution dans le temps. En bref, l'espace de style est la projection quantifiée de propriétés visuelles d'images dans un plan 2D. Chaque dimension du plan représente une caractéristique

visuelle des images numériques (saturation, luminosité, écart type, entropie, etc.). Les images sont alors cartographiées dans l'espace défini par des combinaisons entre ces différentes caractéristiques. Chaque section de l'espace occupée par un groupe d'images est caractérisée par une empreinte qui représente un style particulier. Par exemple, si les caractéristiques visuelles prises en compte sont la luminosité et la saturation, on distinguera des groupes sombres et d'autres lumineux.

Récemment, le même auteur [38] s'est penché sur le problème de la visualisation de grandes collections d'images. La collection étudiée est constituée d'un million de pages de manga. Ces pages correspondent à 883 séries manga disponibles comme "*scanlations*" sur le site OneManga.com [36]. Cette approche est composée de deux étapes. La première étape consiste à extraire les différentes caractéristiques visuelles des images. Puis, une deuxième étape de visualisation permet de présenter l'ensemble des images en les organisant par dimension. Là encore, cette étape rend la visualisation des résultats plus claire pour l'être humain. Elle permet en effet de positionner les images en fonction des caractéristiques visuelles. Chaque ensemble dense des caractéristiques peut alors être identifié comme une classe ou une catégorie de classification. La Figure (2.6) montre par exemple un million de pages de manga affichées selon cette méthode [38] comme des points reflétant la distribution globale des images. Les axes des abscisses et des ordonnées représentent respectivement l'écart type et l'entropie de chaque image.

Cette méthode présente des limites pour la visualisation de grands corpus constitués de millions d'images. Pour étendre la visualisation à des millions d'images, l'auteur a utilisé la technique de nuage de points. La Figure (2.7) montre deux échantillons extraits à partir de la figure précédente. La partie (a) montre la partie supérieure de la visualisation qui contient des pages avec beaucoup de texture et de détails. La

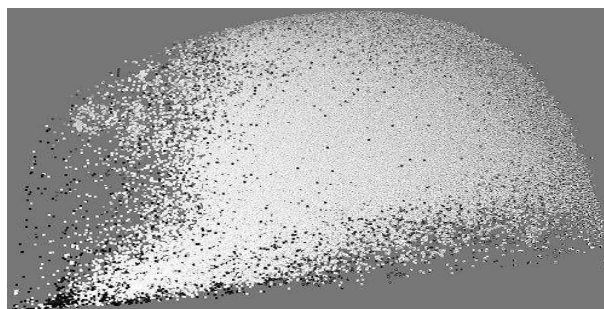


FIGURE 2.6 – Visualisation d’un million d’images sous forme de nuage de points [38]

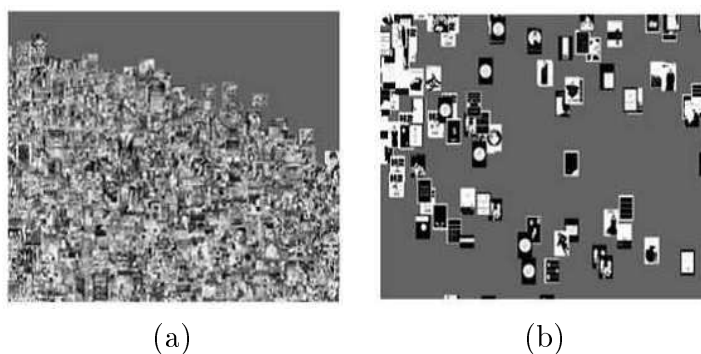


FIGURE 2.7 – Prélèvement de la visualisation d’un million images [38]

partie (b) décrit la partie inférieure droite de la visualisation qui contient des images qui sont soit noires ou blanches ou contenant peu de détails et de texture.

2.2.2 Visualisation par graphes de relations

Parmi les méthodes de visualisation les plus utilisées, on distingue aussi l’utilisation des graphes de relations. Commençons par la définition d’un graphe. Un graphe est l’ensemble de *liens* qui existent entre les éléments appelé des *nœuds* ou sommets. Dans la visualisation par graphes, on peut représenter les *nœuds* par des points et les *liens* par des lignes [48]. La Figure (2.8) montre des exemples de graphes typiques : (a) est un graphe planaire quelconque, (b) est un graphe particulier appelé *un arbre*. Dans un *arbre*, il existe une hiérarchie entre les nœuds, qui est une relation sous la forme *parent-enfant* dans le graphe. Le graphe peut être une forme de visualisation des collections d’images où les images représentent les éléments de cet ensemble et

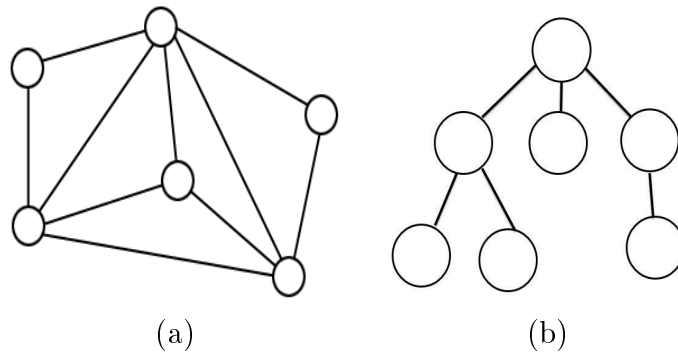


FIGURE 2.8 – Exemple d'un graphe

les liens représentent les relations entre ces derniers [30, 31].

Plusieurs travaux ont utilisé les graphes comme outil pour la visualisation de grandes collections d'images. Selon Herman et al. [31], la façon la plus simple de déterminer l'applicabilité de visualisation de graphes est de considérer la question suivante : "*y a-t-il une relation inhérente entre les éléments de données à visualiser?*". Si la réponse à la question est non, alors les éléments de données sont structurés et le but du système de visualisation de l'information pourrait être d'aider à découvrir les relations entre les données par des moyens visuels. Si, toutefois, la réponse à la question est oui, alors les données peuvent être représentées par les nœuds d'un graphe, avec des liens représentant les relations.

En utilisant les graphes, Wang et al. [61] ont développé l'approche *IMap*. Il s'agit d'une transition animée pour visualiser une collection d'images en fonction de leur similitude visuelle et textuelle sous forme d'un graphe. Comme on le voit sur la Figure (2.9), l'image cherchée ou requête est affichée au centre et le reste des éléments sont organisés selon des rangs de similarité décroissants, selon un motif en spirale. Par défaut, l'image requête est affichée dans une taille normale et la largeur et la hauteur sont réduites de moitié pour les couches externes successives. Zhang et al.[66] ont étudié l'efficacité d'intégration des transitions animées dans la visualisa-

tion des collections d'images. Ils en ont conclu que la tâche d'utilisation de plusieurs étapes de transition animée est plus efficace pour comprendre les relations entre les différents éléments de la collection comparativement à l'absence d'animation [62].



FIGURE 2.9 – Visualisation sous forme spirale [61]

Plus récemment, Gu et al. [30] ont développé l'outil *IGraph* qui visualise sous forme de graphe planaire les images d'une collection et les mots clés utilisés pour leur annotation. Les nœuds du graphe sont constitués par les images et les mots clés. La relation d'une image à l'autre est mesurée par la similarité de leurs caractéristiques visuelles (couleur, spectre de Fourier, etc.). La similarité de deux mots-clés est établie par la co-occurrence des deux mots dans l'annotation de documents textuels indexés par Google. Le graphe peut ainsi aider les utilisateurs à suivre les transitions entre les différentes images et leurs annotations, et aussi permettre la navigation dans des collections massives d'images. L'approche proposée peut générer des graphes composés de millions d'éléments et de centaines des millions de relations entre ces derniers. Pour avoir une exploration efficace, les auteurs ont intégré une représentation graphique progressive et interactive permettant une navigation plus souple sur les images. La Figure (2.10) montre un exemple de cette visualisation en utilisant une base de données d'images astronomique [68]. Les parties (a) et (b) montrent une configuration du graphe avec 110 images et 25 mots clés, et 190 images

et 25 mots clés, respectivement. Les parties (c) et (d) montrent des graphes de 270 images et 50 mots clés, respectivement.

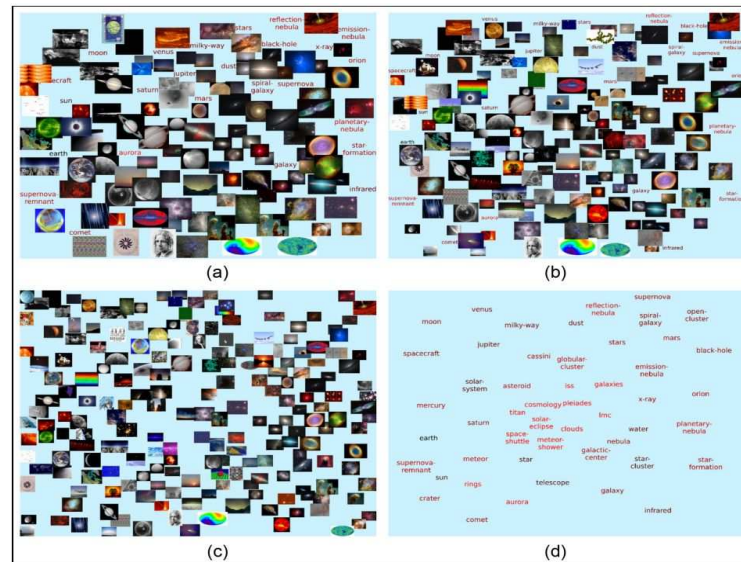


FIGURE 2.10 – Exemple de visualisation avec la méthode IGraph [30].

2.2.3 Visualisation par des grilles 2D/3D

Récemment, d'autres chercheurs ont utilisé un système de grilles (*scénarimage*) afin de visualiser des collections d'images. Un *scénarimage* ou *story-board* est un arrangement de collection d'images dans une grille à deux dimensions, qui permet à un utilisateur de visualiser et de parcourir une collection d'images d'une manière interactive [53]. Cette méthode de visualisation est très utilisée dans les applications de navigation qui permettent aux utilisateurs de mieux parcourir une collection d'images ou de vidéos [54] en offrant :

1. Un aperçu du sommaire du contenu de la collection.
2. La possibilité de trouver le contenu désiré d'une manière interactive.

Sur la base de cette visualisation, Schoeffmann et al. [53] ont développé un algorithme qui permet de visualiser un ensemble d'images triées en fonction de leur



FIGURE 2.11 – Visualisation sous forme d'une grille et cylindre 3D [51]

similarité de couleur. L'algorithme génère un arrangement intuitif des images et affiche les résultats sous plusieurs formats différents (ligne / colonne simple, globe 3D / cylindre, grille, etc.). La Figure (2.11) montre deux interfaces de visualisation de la même collection d'images une sous la forme d'une grille et une autre configuration cylindrique 3D [51].

2.2.4 Autres méthodes de visualisation

Pour faciliter la détection et l'exploration des différentes transitions et relations entre les éléments d'une collection d'images, divers travaux ont abordé ce problème en utilisant d'autres approches que celles mentionnées. La présentation de ces méthodes de visualisation de collections d'images complètera notre revue de littérature. Platt et al. [49] ont développé un outil nommé PhotoToc, qui permet aux utilisateurs de trouver et de visualiser un ensemble de photographies numériques. Ce dernier possède une interface de navigation qui permet de visualiser l'ensemble d'images soit sous la forme d'une vue détaillée de toutes les photos organisées temporellement, soit sous la forme d'une vue d'ensemble de la collection générée automatiquement par un algorithme de classification selon différentes caractéristiques visuelles des images (par exemple la similitude de couleurs).



FIGURE 2.12 – Visualisation basée sur les diagrammes pondérés de Veronoi [18]

Brivio et al. [18] ont proposé une approche de visualisation basée sur les diagrammes pondérés de Veronoi. Les auteurs ont expliqué que la navigation conventionnelle des collections d'images utilise des mécanismes tels que les graphes et les grilles ne sont pas efficaces pour les collections massives. Dans leur travail, ils ont proposé une nouvelle interface basée sur la miniaturisation dont le but est de parcourir et visualiser de grandes collections d'images. La Figure (2.12) montre un exemple de visualisation de collection avec cette approche.

Récemment, de nouvelles approches de visualisation basées sur des techniques d'analyse sémantique d'images ont été proposées [64, 20]. Parmi ces travaux, on peut citer le navigateur baptisé SIB proposé par Yang et al. [64]. Ce navigateur permet non seulement aux utilisateurs de parcourir et de rechercher des images dans les grandes bases de données en fonction du contenu sémantique, mais il permet également aux analystes d'explorer la collection d'images d'une manière visuelle interactive. De plus, il permet d'afficher le contenu d'une grande collection d'images et de faire des recherches interactives (recherche par contenu, par similitude de caractéristiques, etc.). La Figure (2.13) montre les corrélations entre l'image centrale de fond et le reste des images de la collection, qui sont indiquées par les distances verticales entre elles. On peut constater qu'il existe un groupe d'images contenant la neige et les



FIGURE 2.13 – Visualisation basée sur l'analyse sémantique [64]

montagnes qui sont similaires à l'image requête qui est encadrée par la couleur rose. Les images sélectionnées qui ressemblent à la requête sont encadrées en vert.

2.3 Limites de la visualisation du corpus visuel

Les travaux cités précédemment ont été pour la plupart proposés pour traiter des grandes collections d'images à des fins de classification et/ou de visualisation. Bien qu'ils soient efficaces pour classer et visualiser des images en groupes homogènes à partir des caractéristiques visuelles de bas niveau (couleur, texture, entropie, etc.), ils ne prennent pas en compte l'ordonnement temporel des images recherchées sur le Web. En effet, dans plusieurs domaines, tels que l'analyse des tendances de groupes artistiques, politiques, ethniques, culturels, scientifiques ou autres, il est important de comprendre les éléments visuels partagés par les groupes et de suivre leur évolution dans le temps.

Les outils d'analyse d'images utilisés par les méthodes de recherche/visualisation existants sont limités pour le retracement de l'évolution des images et l'analyse des relations entre, par exemple, une image originale et ses différentes variantes existantes sur le Web. En d'autres termes, bien que ces outils permettent de regrouper les images les plus similaires, ils ne sont pas capables de fournir la relation (notam-

ment chronologique) entre les images d'un corpus. En résumé, ces méthodes ne sont pas adaptées pour :

- Détecter le type de transformation appliquée à une image originale afin de donner l'image dérivée (déterminer si l'image est recadrée ou compressée, etc.).
- Détecter le type de dégradation subie par une image originale et observer une de ces variantes (ex. déterminer si l'image est recadrée ou compressée, etc.).
- Suivre l'évolution temporelle, au sens de détecter les différents dérivés et transformations d'une image depuis leur création jusqu'à une version dérivée.

Les besoins d'analyse des relations entre les éléments d'un corpus visuel vont au delà de l'analyse simple de la similitude des caractéristiques visuelles entre images. Non seulement il faudra utiliser la similitude entre les images, mais ils requièrent aussi l'utilisation d'outils permettant d'inférer l'historique d'évolution et les types de modifications effectuées sur une image. Ceci est important pour extraire des informations telles que l'originalité des images, et pour retracer l'histoire d'utilisation des images sur le Web.

2.4 Sommaire du chapitre

Dans ce chapitre, nous avons présenté des travaux existants traitant de la recherche d'images basée sur le contenu, ainsi que la visualisation de collections d'images. Nous avons présenté diverses approches de recherche utilisant les caractéristiques visuelles, textuelles et la combinaison des deux. Nous avons aussi présenté quelques techniques existantes pour la visualisation de collections d'images permettant de faciliter la navigation et l'étude de la similarité entre images. Dans le chapitre suivant, nous allons présenter une revue de littérature sur les différentes transformations d'une image numérique et les différents standards de métadonnées.

Chapitre 3

Caractéristiques de transformations d'images

3.1 Introduction

De nos jours, avec la puissance et l'évolution des logiciels d'édition et de traitement d'images (tels que Photoshop, Paint Shop Pro, etc.), il est devenu très facile d'apporter des modifications à des images ou à des photographies personnelles. Dans ce chapitre, nous allons d'abord faire un survol des types de transformations *photométriques* et *géométriques* qu'on peut appliquer sur une image numérique. Par la suite, nous présenterons les différents standards de *métadonnées* décrivant les images numériques. Le but de catégoriser les types de transformations d'images est de nous permettre par la suite de fonder les outils qui nous serviront à détecter les types de transformations appliquées sur les images, de retrouver les images originales d'un groupe d'images similaires ayant la même origine et de retracer l'historique d'évolution d'une image en analysant les changements apportés à cette dernière dans ses autres versions existantes sur le Web.

3.2 Transformations photométriques

Dans le domaine de l'imagerie, le terme *transformation photométrique* réfère à la catégorie de manipulation d'images où les valeurs originales des pixels sont modifiées, mais la géométrie (c'est-à-dire la disposition spatiale des pixels) de l'image reste inchangée. En effet, ces types de transformations permettent de changer les valeurs de niveau de gris ou de couleur des pixels dans une image, sans influencer la relation spatiale entre ces derniers. Voici une liste non-exhaustive des différentes transformations photométriques possibles pour une image numérique.

3.2.1 Transformations du niveau de gris

Dans la suite du chapitre, les valeurs de pixels avant et après traitement seront désignées par r et s , respectivement. Ces valeurs sont liées par une expression de la forme $s = T(r)$, où T est une transformation photométrique qui fait correspondre une valeur de pixel r en une valeur de pixel s [1]. Généralement, les niveaux de gris sont des nombres naturels situés dans l'intervalle $[0, L-1]$ où L est le nombre total de niveaux de gris. Nous allons commencer par les trois fonctions de base d'amélioration d'images.

a) Négatif d'une image

La modification des valeurs d'intensité d'une image des niveaux de gris dans l'intervalle $[0, L-1]$ selon l'équation suivante produit l'équivalent d'un négatif photographique :

$$s = L - 1 - r \quad (3.1)$$

Ce type de traitement est particulièrement utilisé dans le but d'améliorer et de rendre plus visible les détails de régions sombres d'une image, surtout lorsque ces zones sont dominantes. La Figure (3.1) montre un exemple de cette transformation. En dépit du fait que le contenu visuel est le même dans les deux images. On remarque qu'il

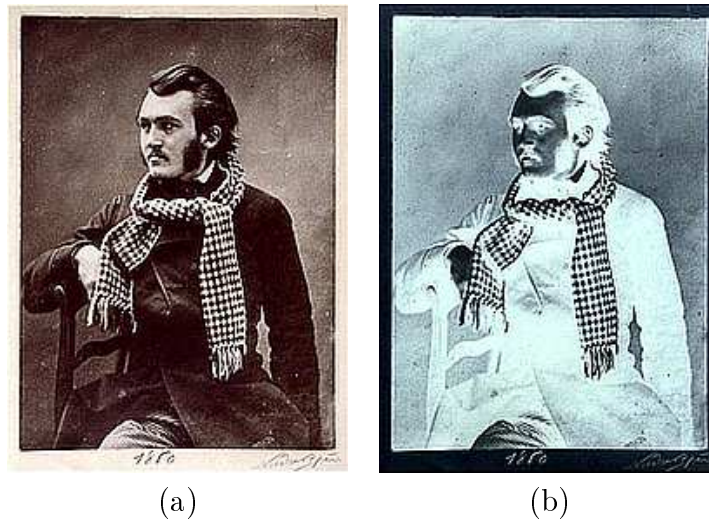


FIGURE 3.1 – Exemple du négatif d'une image : (a) image originale, (b) son négatif. est plus facile d'analyser et de visualiser les régions sombres de l'image originale (b).

b) Transformation logarithmique

Cette transformation est utilisée pour illuminer des valeurs sombres de pixels dans une image tout en maintenant les valeurs claires inchangées. La formule de transformation logarithmique est donnée par l'expression :

$$s = c \log(1 + r), \quad (3.2)$$

où c est une constante et r et s représentent les niveaux de gris original et transformé du pixel, respectivement. La Figure (3.2) illustre ce type de transformation.

c) Correction Gamma

Cette transformation permet d'augmenter ou de diminuer la luminosité globale d'une image. Elle est formulée comme suit :

$$s = r^\gamma \quad (3.3)$$

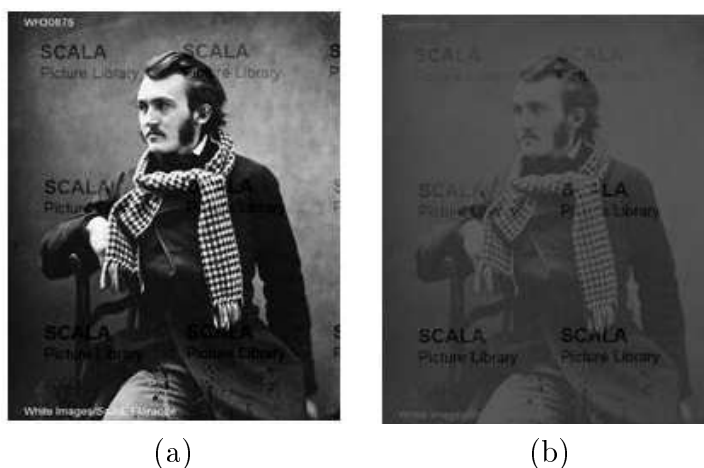


FIGURE 3.2 – Exemple de transformation logarithmique d'une image : (a) image originale, (b) image transformée



FIGURE 3.3 – Exemples de correction Gamma : a) image originale, b) $\gamma = 0.4$ et c) $\gamma = 2$

où $\gamma > 0$ est un paramètre contrôlant le type de changement de luminosité. Lorsque $\gamma < 1$, l'image résultat sera plus illuminée que l'image originale. Lorsque $\gamma > 1$, l'image résultat sera plus sombre que l'image originale. Noter que pour appliquer cette transformation, il faudra d'abord normaliser les valeurs de niveau de gris de l'image originale en les divisant par $L - 1$. La Figure (3.3) montre des exemples de corrections Gamma.

d) Égalisation d'histogrammes

L'histogramme représente la distribution des niveaux de gris dans une image, c'est-à-dire le nombre de pixels de l'image pour chaque niveau de gris r . La Figure (3.4) montre une illustration d'un histogramme où l'axe des abscisses représente le niveau de gris r (de 0 à 255) et l'axe des ordonnées représente le nombre de pixels ayant niveau de gris r .

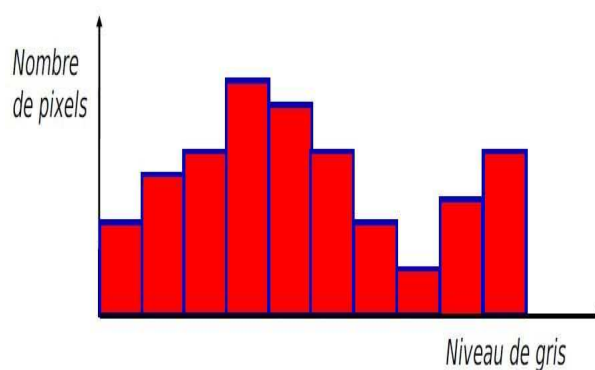


FIGURE 3.4 – Exemple d'un histogramme

Les niveaux de gris sombres apparaissent à gauche, et les niveaux de gris clairs à droite de l'histogramme. L'histogramme est une technique utilisée pour étudier la répartition statistique globale des niveaux de gris. L'un de ces traitements est *l'égalisation d'histogramme*, qui permet de modifier la luminosité de l'image pour la rendre plus uniforme, et ceci sans altérer la structure visuelle des pixels. Elle permet aussi d'harmoniser la répartition des niveaux de luminosité de l'image de sorte que les fréquences des niveaux de gris soient proches. La Figure (3.5) montre un exemple d'une égalisation d'histogramme. Les niveaux de gris de l'image égalisée seront représentés par le même nombre de pixels dans le nouvel histogramme. Comme le montre la Figure (3.5), on distingue parfaitement les différents objets constituant l'image, et même certains détails invisibles avant l'égalisation.

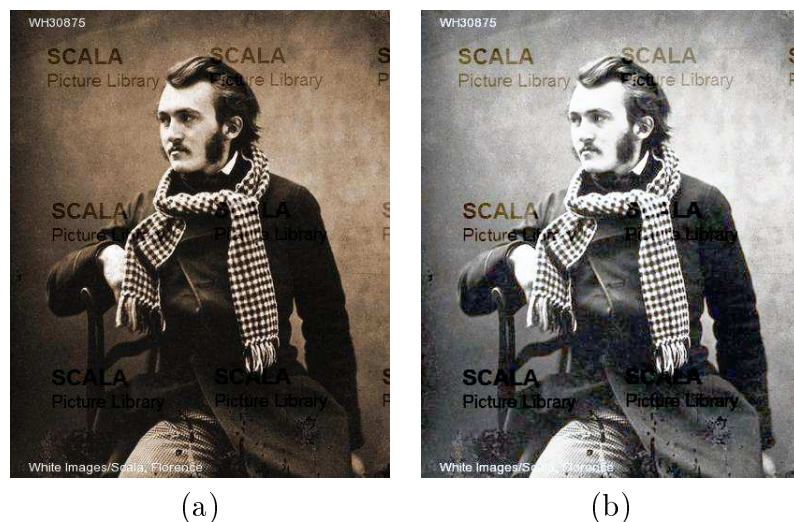


FIGURE 3.5 – Exemple de transformation par égalisation d’histogramme : (a) image originale, (b) image transformée

e) Amélioration du contraste

Le contraste d’une image est lié à la notion de luminosité. Une image paraît contrastée lorsqu’elle contient des zones très lumineuses et d’autres très sombres, et ainsi les détails de l’image paraissent plus nets. L’image est moins contrastée si elle est très homogène. La Figure (3.6) montre un exemple illustrant l’amélioration de contraste d’une image. La partie (a) est caractérisée par un faible contraste, son histogramme est très étroit et la luminosité très homogène. Au contraire, la partie (b) est plus contrastée et les détails sont mieux apparents.

3.2.2 Filtrage et réduction de bruit

Au cours de leur acquisition et transmission, les images peuvent subir des dégradations liées au bruit, qui est une information non pertinente ou utile rajoutée à l’image. Lors de l’acquisition d’une image, la performance des cellules photosensibles est généralement affectée par différents facteurs (ex. conditions atmosphériques, qualité du capteur, etc.). Lors de la transmission, les images peuvent être corrompues à cause d’interférences dans le canal de transmission. Le bruit dans une image numé-

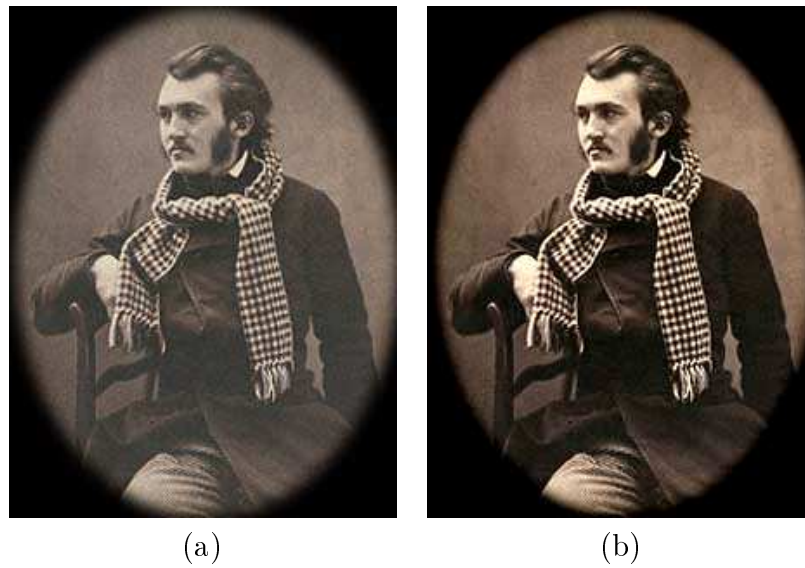


FIGURE 3.6 – Exemple d'amélioration de contraste : (a) image originale, (b) image dont le contraste a été amélioré

rique se manifeste par la présence de pixels parasites dégradant la qualité de l'image et peut affecter la *luminance* et la *couleur* de l'image, selon les modalités décrites ci-dessous.

- **Effet sur la luminance** : Modifie la structure des pixels et donne une texture granuleuse à l'image ;
- **Effet sur la chrominance** : Agit sur la couleur des pixels de l'image par l'ajout de bruit aux bandes de couleurs.

L'objectif de la restauration d'images est d'utiliser certains traitements sur l'image bruitée afin d'obtenir une image proche de sa version originale. Il existe plusieurs techniques pour réduire le bruit présent dans une image et produire une image plus nette. La plupart de ces dernières se basent sur le filtrage d'images [1]. La Figure (3.7) illustre une restauration d'image bruitée par un filtre Gaussien [1] :



FIGURE 3.7 – Exemple de réduction de bruit d'une image : (a) image originale, (b) image bruitée, c) réduction de bruit par filtre Gaussien de taille 5×5



FIGURE 3.8 – Exemple de *morphing* entre deux images (celle de droite et de gauche). Le résultat du *morphing* est l'image du milieu (Ref. Wikipedia)

3.2.3 Morphose (*morphing*)

Un autre type de transformation photométrique qu'on peut appliquer sur une image numérique est le morphose (*morphing*). Il consiste à fabriquer une animation qui permet de transformer progressivement une image initiale en une autre image finale de la manière la plus fluide possible [63, 35]. Cette transformation est basée sur la connaissance d'un certain nombre de points de référence qui devront être fixés sur l'image de départ et d'arrivée (position des yeux, des oreilles et de la bouche, etc.). Ces points sont utilisés pour calculer les coefficients appropriés pour la déformation. La Figure (3.8) montre un exemple de cette transformation.

RGB vers CMY	CMY vers RGB
Cyan = 1 - Red	Red = 1 - Cyan
Magenta = 1- Green	Green = 1- Magenta
Yellow = 1- Bleu	Blue= 1- Yellow

TABLE 3.1 – Transformations entre les espaces RGB et CMY

3.2.4 Conversion du mode colorimétrique d'une image

En général, on représente la couleur à l'aide de trois valeurs (espace en trois dimensions). Il existe plusieurs espaces de couleurs :

- RGB (red, green, blue) : pour moniteurs et caméras ;
- CMY et CMYK (cyan, magenta et yellow) : pour les imprimantes ;
- HSI (hue, saturation et intensity) : plus près de l'interprétation humaine car cet espace sépare la teinte de la luminosité de l'image.

a) L'espace RGB

L'espace de couleurs RGB est le plus répandu. En effet, il est utilisé dans la majorité des outils matériels de visualisation. Dans cet espace, un pixel est codé par trois composantes : rouge, vert et bleu. Cet espace a été développé en fonction des connaissances liées à la vision humaine [32]. Ce modèle est dit *additif*, ce qui signifie que toutes les couleurs possibles sont déduites à partir de l'addition des composantes R, G et B [1].

b) L'espace CMY

CMY est un modèle dit *soustractif* basé sur les couleurs complémentaires à l'égard de couleur du modèle de couleur RGB [32]. Généralement, cet espace est utilisé pour les périphériques de sortie tels que les imprimantes. Le tableau (3.1) illustre les différentes équations de transformations entre les deux espaces RGB et CMY.

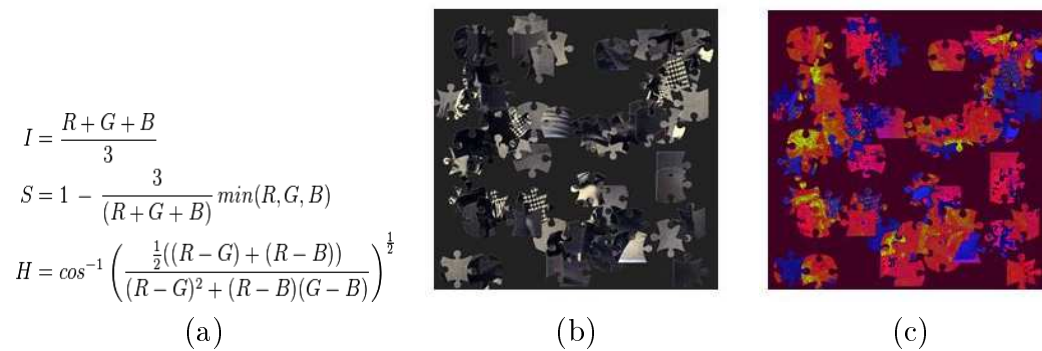


FIGURE 3.9 – Exemple de transformation $RGB \rightarrow HSI$: a) formule de transformation $RGB \rightarrow HSI$, (b) image dans l'espace RGB, (c) image dans l'espace HSI

c) L'espace HSI

Le modèle de couleur HSI est basé sur le système visuel humain. Cet espace utilise les coordonnées cylindriques pour représenter les points RGB. La composante (H : hue) mesure la teinte et la pureté des couleurs, la composante (S : saturation) indique le degré de couleur blanche mélangée avec une couleur spécifique, et (I : illumination) se réfère à l'intensité de la lumière. Le passage d'un mode colorimétrique à un autre présente un autre type des transformations photométriques des images. Par exemple, la conversion entre les deux espaces RGB et HSI est illustrée dans la Figure (3.9).

d) Conversion de la couleur au niveau de gris

On peut construire une image en niveaux de gris à partir d'une image couleur en calculant la moyenne des trois canaux (rouge R, vert V et bleu B). Autrement dit, pour représenter un niveau de gris dans une image couleur, il suffit d'associer la même valeur pour les trois composantes. On calcule donc une valeur A , qui s'appelle la luminance de la couleur $A = \frac{(R+V+B)}{3}$. La Figure (3.10) montre un exemple de conversion en niveau de gris d'une image couleur.

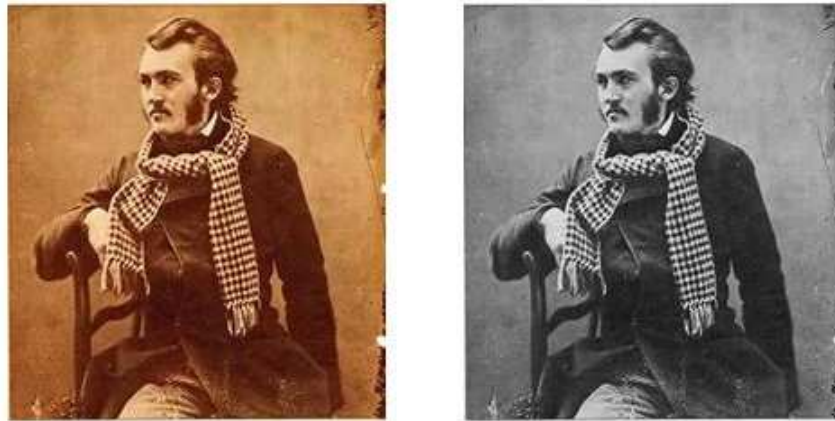


FIGURE 3.10 – Exemple de conversion d'une image couleur en niveau de gris

3.2.5 Édition d'images

a) Recadrage (*cropping*) d'images

Le recadrage d'une image numérique est un traitement qui sert à enlever une partie de cette dernière. Cela peut impliquer de couper une partie de l'image du côté gauche, droite, haut, bas ou une combinaison de ces derniers. Le recadrage peut se manifester simplement par l'enlèvement d'un petit rectangle de l'image. Cette partie découpée de l'image requiert de fournir une paire de coordonnées (x_1, y_1) et (x_2, y_2) qui définissent les coins de la nouvelle image recadrée. La Figure (3.11) montre un exemple de recadrage où le visage est extrait à partir d'une image de notre corpus.

b) Insertion d'un texte ou d'un objet dans l'image

Un autre type de transformation photométrique que l'on peut appliquer sur les images numériques est l'insertion d'un texte ou d'un objet dans l'image. La Figure (3.12) montre un exemple d'insertion de texte dans une image du corpus étudié.

c) Insertion de bande (*image padding*)

Ce type de transformation peut être effectuée avec deux types de traitement :

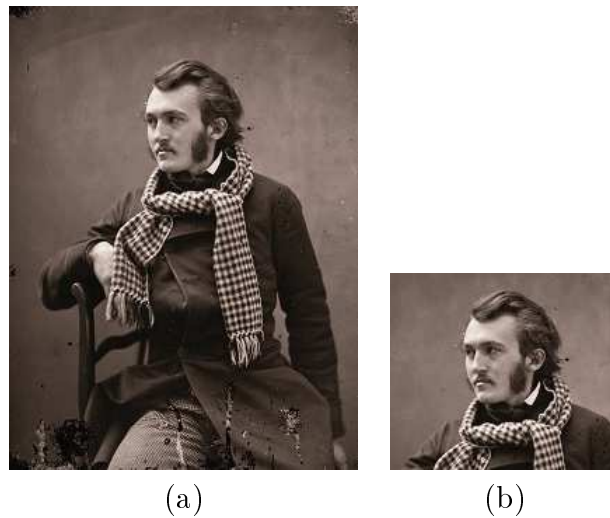


FIGURE 3.11 – Exemple de *cropping* d'image : a) image originale, b) image croppée



FIGURE 3.12 – Exemple d'insertion de texte dans une image

- **Le rembourrage (padding)** : défini par l'insertion ou l'ajout de nouveaux pixels autour des bords d'une image (c'est-à-dire un rembourrage autour du contenu). Comme montre la Figure (3.13), un rembourrage peut être appliqué sur tous les côtés (haut, bas, droit et gauche). Dans ce cas, la taille de l'image originale doit être changée.
- **Insertion d'une bordure** : contrairement au rembourrage, dans ce type de traitement, la bordure ou la bande chevauche l'image, comme le montre l'exemple de la Figure (3.14).

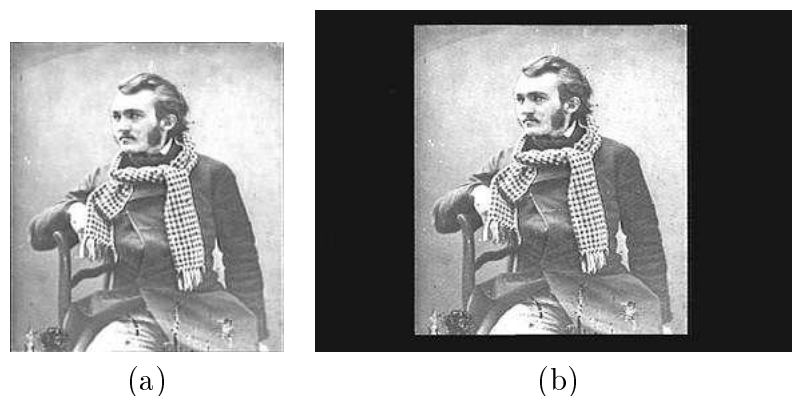


FIGURE 3.13 – Exemple de rembourrage d'une image : a) image originale, b) image rembourrée

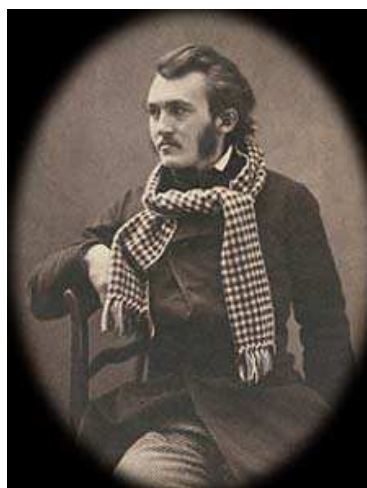


FIGURE 3.14 – Exemple d'insertion de bordure

3.3 Transformations géométriques de l'image

Le terme *transformation géométrique* se réfère à la classe des transformations d'images où la géométrie de l'image est modifiée, mais les valeurs des niveaux de gris ou des couleurs de pixels restent inchangées. Autrement dit, les transformations géométriques permettent de modifier la relation spatiale entre les pixels, comme illustré sur la Figure (3.15). Dans une transformation spatiale T d'une image, chaque point (x, y) est déplacé dans une autre position spatiale (u, v) via une formule générique écrite comme suit :

$$[(u, v) = T(x, y)] \quad (3.4)$$

Avec (x, y) représentant les coordonnées des pixels dans l'image originale et (u, v) représentant les coordonnées des pixels dans l'image transformée [1].

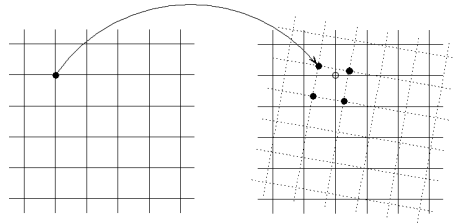


FIGURE 3.15 – Transformation des coordonnées de pixels

3.3.1 Transformations affines

L'une des transformations spatiales de coordonnées les plus courantes est la transformée affine qui a la formulation générale suivante :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.5)$$

Le tableau (3.2) résume la liste des transformations existantes.

- **Translation** : consiste à simplement déplacer l'image horizontalement et/ou verticalement avec une valeur t_x sur l'axe des x et t_y selon l'axe des y .
- **Changement d'échelle** : consiste à changer l'échelle d'une image dans la direction verticale ou horizontale ou les deux à la fois.
- **Rotation** : consiste à tourner l'image sur son centre avec un angle θ . Chaque pixel (x, y) est alors déplacé avec la rotation.
- **Cisaillement** : consiste à amplifier graduellement une coordonnée en fonction de l'autre. Cela a un effet de cisaillement sur la forme générale de l'image et de son contenu.






Nom de Transformation	Matrice de transformation	Exemple
Changement d'échelle	$\begin{pmatrix} c_x & 0 & 0 \\ 1 & c_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$	
Rotation	$\begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$	
Translation	$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$	
Cisaillement horizontal	$\begin{pmatrix} 1 & s_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	
Cisaillement vertical	$\begin{pmatrix} 1 & 0 & 0 \\ s_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	

TABLE 3.2 – Transformations géométriques affines d'une image [1].

3.3.2 Transformations non-linéaires

a) Tournoiement

Le tournoiement est une sorte de rotation dont l'angle de rotation est proportionnel à la distance radiale de l'image. Cette transformation est inspirée de la transformation en coordonnées polaires. L'angle de rotation θ est désormais défini comme :

$$\theta = \arctan(\Delta_x/\Delta_y) + \phi \frac{r_{max} - r}{r_{max}}, \quad (3.6)$$

où Δ_x et Δ_y sont les différences des coordonnées x et y du pixel avec le centre de l'image (x_c, y_c) , et r est la distance radiale du pixel. L'effet de la transformation est que la partie centrale reste à la même position et le reste des pixels est mis en rotation autour du centre avec un angle de rotation qui est au maximum de ϕ degrés. Cette transformation est définie comme suit :

$$\begin{cases} x = x_c + r \cos(\theta) \\ y = y_c + r \sin(\theta) \end{cases} \quad (3.7)$$

Pour chaque pixel (x_1, y_1) dans l'image de sortie, on calcule une position de pixel dans l'image d'entrée (x, y) . Si ce dernier n'est pas inclus dans l'image d'entrée, le pixel (x_1, y_1) en sortie sera mis au noir. La Figure (3.16) montre un exemple d'une transformation de tournoiement avec deux différentes valeurs de l'angle de rotation ϕ .

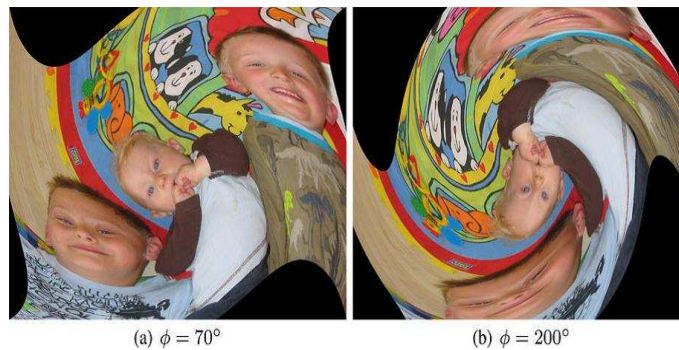


FIGURE 3.16 – Exemples de tournoiement [1]

b) Transformation sphérique

Cette transformation est appliquée directement sur le centre de l'image. Elle permet de zoomer sur le centre, la taille de la zone agrandie étant définie par la variable S . L'effet de zoom est similaire à celui d'une lentille de caméra [1]. Cette transformation est exprimée par l'équation suivante :

$$\begin{cases} x = x' - t \tan(\alpha_x) \\ y = y' - t \tan(\alpha_y) \end{cases} \quad (3.8)$$

Où S et n sont définis par l'utilisateur, et x et y sont tels que définis ci-dessus. Un exemple d'une transformation sphérique est illustré dans la Figure (3.17).

(a) $S = 600$ and $n = 1.8$ (b) $S = 300$ and $n = 2.5$

FIGURE 3.17 – Exemple d'une transformation sphérique [1].

c) Transformation d'ondulation

La transformation d'ondulation déforme l'image localement en utilisant une fonction de *sinus*. La forme générale de cette transformation est la suivante :

$$\begin{cases} x = x' + a_x \sin((y' 2\pi)/w_x) \\ y = y' + a_y \sin((x' 2\pi)/w_y) \end{cases} \quad (3.9)$$

Où a_x et a_y sont les amplitudes de la forme d'onde sur les directions x et y , respectivement, et w_x et w_y ont le rôle de contrôler les fréquences des ondes sur l'axe des x et des y , respectivement. Un exemple de transformation d'ondulation est illustré dans la Figure (3.18).

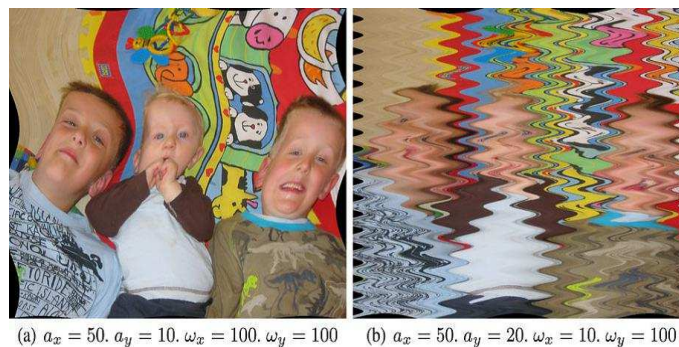
(a) $a_x = 50$, $a_y = 10$, $w_x = 100$, $w_y = 100$ (b) $a_x = 50$, $a_y = 20$, $w_x = 10$, $w_y = 100$

FIGURE 3.18 – Exemple d'une transformation d'ondulation [1]

d) Transformation locale

Dans les quatre transformations géométriques ci-dessus, tous les pixels passent par le même type de transformation. Mais il est possible d'appliquer différentes transformations localement. Évidemment, cela peut entraîner de nombreuses sorties différentes en combinant les différentes transformations représentées ci-dessus. Un exemple de cette transformation basée sur la rotation est présenté à la Figure (3.19). L'image d'entrée a été d'abord copiée à partir de l'image de sortie afin d'éviter des pixels vides dans la sortie. Ensuite, l'image d'entrée est divisée en un certain nombre de carrés ayant chacun la taille $S \times S$. Chaque carré dans l'entrée est alors tourné et mis en correspondance avec l'image de sortie. L'angle de rotation est soit θ ou $-\theta$, en fonction de sa position d'entrée. Autrement dit, le premier carré est tourné de θ degrés, le deuxième avec $-\theta$ degrés, le troisième est tourné avec θ et ainsi de suite.

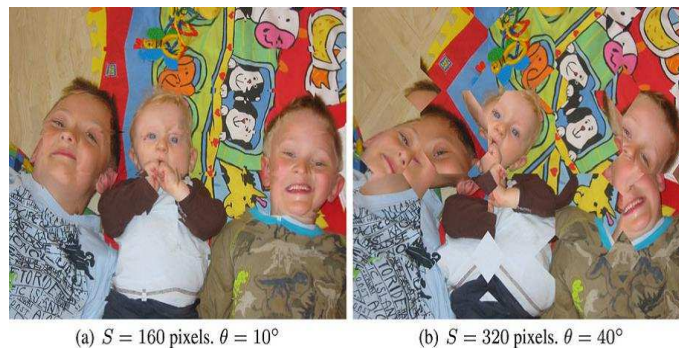


FIGURE 3.19 – Transformation locale avec rotation

3.4 Les métadonnées d'une image numérique

3.4.1 Définition et intérêts

Les métadonnées sont des informations textuelles et numériques utilisées pour décrire et indexer un fichier numérique dans une collection [45]. Dans cette section, nous allons présenter les différents standards de métadonnées d'une image numérique ainsi que leurs utilisations et modes d'exploitation. Par définition, une métadonnée

est un ensemble d'informations structurées permettant de décrire une ressource, qui est une image dans notre cas. Il existe deux manières de saisir ces informations dans le fichier image : soit manuellement par l'intervention d'utilisateurs, soit de manière automatique via les appareils photo numériques lors de la prise de vue images. Ces métadonnées peuvent être stockées de deux manières différentes :

- **En interne** : incluses dans le fichier de l'image. Ce type de stockage est applicable sur la plupart des formats d'images.
- **en externe** : en dehors du fichier d'image (nom de l'image, sa taille sur le disque, etc.).

Dans le cadre de notre projet visant à suivre l'évolution des images, il est très important et utile que les métadonnées décrivant une image restent disponibles même si cette dernière a subi plusieurs traitements. En effet, les métadonnées sont très utilisées dans diverses applications telles que l'identification et la protection des droits d'auteurs. Elles sont également utiles pour la gestion des images afin de trouver et d'indexer facilement des objets numériques dans les collections (par date, nom du créateur, lieu, etc.). Finalement, certaines métadonnées permettent de suivre l'historique des différents traitements appliqués sur l'image (ex. création, modification, etc.). Dans la plupart des cas, l'image garde l'ensemble de ces métadonnées lorsqu'elle est traitée (ex. téléchargement, copie, compression, etc.).

3.4.2 Différents standards

Dans le domaine de la photométrie, il existe trois principaux standards de métadonnées internes : **EXIF**, **IPTC** et **XMP**.

a) Métadonnées **EXIF**

Il s'agit d'un format de métadonnées produit par les caméras numériques afin de décrire les photos. Ce format a été développé en 1995 par des industriels Japonais.

Les métadonnées EXIF sont générées de manière automatique par l'appareil photo lors de la capture d'une image. La plupart des champs EXIF sont techniques (dimensions, taille, résolution), mais ils contiennent aussi des informations temporelles et spatiales concernant les conditions de la prise de vue (la date, heure, localisation GPS, etc.). De plus, ces métadonnées peuvent être traitées par des logiciels de visualisation et de traitement d'images. Ces outils conservent généralement les données EXIF lors des traitements successifs sur les images. La Figure (3.20) montre une partie des métadonnées EXIF d'une image de notre corpus visualisées par l'outil *Exifviewer*.



File Type	JPEG
MIME Type	image/jpeg
Comment	(253 bytes binary data)
Encoding Process	Baseline DCT, Huffman coding
Bits Per Sample	8
Color Components	3
File Size	44 kB
File Type Extension	jpg
Image Size	374 × 477
Y Cb Cr Sub Sampling	YCbCr4:2:0 (2:2)

FIGURE 3.20 – Exemple de métadonnées EXIF générées avec ExifViewer

b) Métadonnées IPTC

Le standard IPTC est une structure de métadonnées applicable à plusieurs formats de fichiers (texte, image, etc.). En ce qui concerne les images, contrairement au format EXIF, les attributs prévus par le modèle IPTC sont destinés à la description

du contenu visuel de l'image. De plus, ils sont décrits manuellement, ce qui nécessite l'intervention humaine. Ils reprennent quelques champs EXIF de base (créateur de l'image, titre, date, localisation géographique) auxquels s'ajoutent des éléments décrivant le contenu de l'image (mots-clés, légende, description détaillée, crédits, conditions d'utilisation). La première intégration de ce standard de métadonnées dans les images a été réalisée par les logiciels d'Adobe dans les années 1990, ce qui lui a valu un large succès. En effet, ce standard et ce type de métadonnées sont très reconnus et utilisés dans plusieurs domaines professionnels de l'image numérique (agences photos, journaux, etc.) [59, 23] à des fins de classification et recherche d'images, ainsi que pour la gestion des droits d'auteur.

c) Métadonnées XMP

XMP (Extensible Meta data Platform), comme son nom l'indique, est un standard conçu pour fournir une plateforme dont le but est d'assurer l'extensibilité des métadonnées pour diverses applications. Ce standard, introduit par Adobe en 2001, est fondé sur le langage à balises XML, ainsi que sur le schéma de métadonnées du *Dublin Core* et sur une version simplifiée de RDF, qui permet d'intégrer des métadonnées dans plusieurs formats de fichiers (TIFF, JPEG, JPEG 2000, PDF, PNG, HTML, etc.).

- **XML** : langage de balisage extensible recommandé par le consortium W3C en 1995 [12]. Grâce à son extensibilité, XML peut décrire n'importe quel domaine de données.
- **RDF** : standard lancé en 1999 et finalisé par le W3C en 2004. Le but de standard RDF est de décrire les ressources Web et leurs métadonnées de manière formelle [59]. En effet, c'est un moyen d'encoder des métadonnées structurées avec le langage XML.
- **Dublin Core** : initiative développée en 1995 lors d'une conférence tenue à Dublin, qui est devenue un standard depuis 2003. Il s'agit d'un jeu de méta-

données qui permet de décrire les documents sur le Web ou d'autres ressources documentaires. Le schéma de Dublin Core comprend quinze éléments de base, avec un seul niveau d'arborescence sans raffinement d'éléments. Ces éléments sont répartis autour de trois domaines décrivant les ressources documentaires :

1. **Contenu** : titre, sujet, description, source, langue, relation, couverture.
2. **Propriété intellectuelle** : créateur, éditeur, contributeur, droits.
3. **Instance particulière** : date, type, format, identifiant.

3.4.3 Les outils de gestion des métadonnées d'images

Les outils de traitement des métadonnées d'une image numérique sont nombreux. Dans cette partie nous allons décrire l'outil *Exiftool* qui est considéré comme l'un des logiciels les plus puissants permettant de manipuler les métadonnées EXIF, IPTC et XMP. Ce logiciel est composé d'une librairie Perl et d'une application de ligne de commande qui permet de traiter les métadonnées d'images de différentes manières (lire, écrire, modifier, supprimer, etc.). De plus, ce dernier supporte plusieurs formats d'images (DNG, GIF, JPEG, PDF, PNG, etc.). Finalement, ce logiciel est gratuit et peut fonctionner sur différentes plateformes (Windows, Unix, Mac). Voici une liste des quelques caractéristiques de ce logiciel :

- Puissant, rapide, flexible.
- Prend en charge un grand nombre de formats de fichiers différents.
- Supporte plusieurs types de métadonnées.
- Possibilité d'ajouter des annotations définies par l'utilisateur.

3.4.4 Problèmes liés aux outils de gestion de métadonnées

Les outils de gestion et de traitement d'images sont nombreux. Certains permettent la gestion et la préservation des métadonnées, mais d'autres peuvent in-

roduire des erreurs, voici une liste des quelques problèmes qu'il est possible de rencontrer :

1. L'outil très reconnu *ImageMagick*, qui permet notamment de créer ou de convertir des images en différents formats, ne supporte que les métadonnées IPTC et EXIF. Donc si les images manipulées par ce logiciel contiennent de métadonnées XMP, elles seront perdues.
2. Même un logiciel comme Photoshop a pu présenter des problèmes (par exemple, dans certains cas, faire disparaître les métadonnées des images).
3. la librairie GD, destinée à la création d'images ne conserve aucune métadonnée.

3.5 Sommaire du chapitre

Dans ce chapitre, nous avons présenté les différentes transformations géométriques et photométriques des images, ainsi que les différents standards et types des métadonnées pour une image numérique. Dans le chapitre suivant, nous allons faire usage de quelques unes de ces transformations et de leurs caractéristiques afin de développer notre méthodologie pour quantifier les relations entre les images et suivre l'évolution temporelle de ces dernières.

Chapitre 4

Méthode proposée et discussion des résultats

4.1 Introduction

L'objectif principal de notre mémoire est d'élaborer une méthodologie pour analyser un corpus d'images afin de déterminer les relations de similarité entre les copies images et inférer l'ordre chronologique de leur créations/mise en ligne sur le Web. Ce chapitre décrit notre méthodologie générale proposée pour atteindre cet objectif, en mettant l'accent sur les différentes techniques utilisées. Il contient aussi des résultats préliminaires obtenus sur le corpus étudié, en indiquant les avantages et les limites de notre solution. À terme, notre méthodologie mènera au développement d'algorithmes capables de :

- détecter les différents types de transformations géométriques et photométriques appliquées entre les images originales et les différentes copies afin d'estimer les relations de précedence. Ceci mènera à un triage des images selon un ordre chronologique en se basant sur les diverses transformations détectées.
- Identifier et visualiser les différentes lignées dans un corpus d'images constituées d'images originales et leurs copies ou versions modifiées. De manière plus

explicite, une *lignée* sera toute branche constituée par les différentes variantes ou copies de la même image racine. Par ailleurs, l'ensemble des lignées formera un *arbre de précedence*.

- Annoter l'arbre de précedence en identifiant les transformations entre couple d'images voisines. En particulier, une approche probabiliste est proposée pour catégoriser les transformations d'édicions.

4.2 Méthodologie proposée

Nous présentons dans la Figure (4.1) les étapes principales de notre méthodologie pour atteindre les objectifs mentionnés. Le point d'entrée de la méthode proposée est la réception du corpus à étudier, et possiblement d'un ensemble d'images racines. Le point de sortie est la production d'un graphe arborescent qui permettra la classification des images selon un ordre temporel. Le graphe est construit grâce à la détection des différentes relations de précedence, ainsi qu'à l'analyse des différentes transformations entre les différentes copies d'images. Dans la section suivante, nous expliquons ces étapes en commençant par le processus de prétraitement requis pour les images.

On notera que, puisque nos données à classifier sont des images et qu'avec Matlab on ne peut pas générer des graphes où les nœuds sont des images (voir Figure 4.5), nous avons utilisé la bibliothèque JavaScript **D3.js** pour visualiser les graphes des lignées d'images. Cette dernière permet de lier des données arbitraires à un *DOM* (Document Object Model), puis appliquer des transformations basées sur les données. Par exemple, on peut utiliser D3 pour générer un tableau HTML à partir d'un tableau de nombres, pour créer un graphique SVG interactif sous forme de barres ou d'un graphe hiérarchique (arbre) avec transitions et interactions.

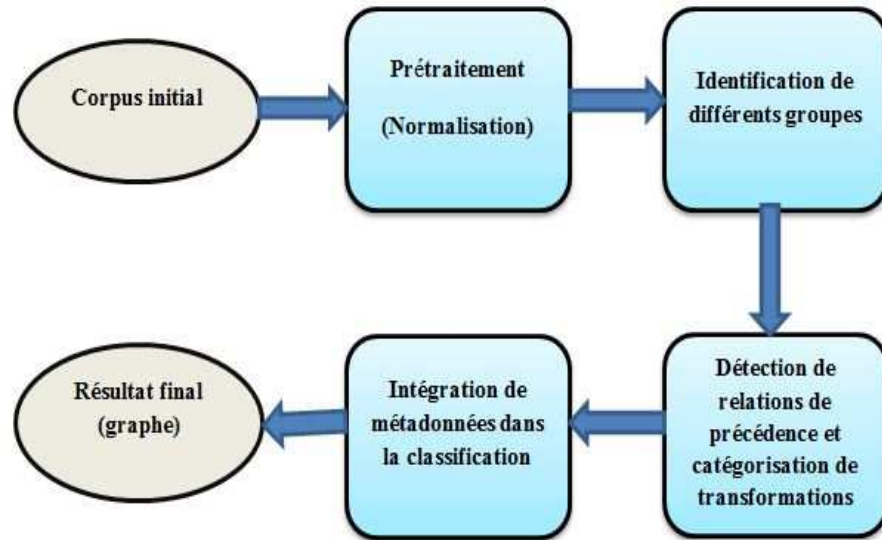


FIGURE 4.1 – Étapes de la méthode proposée

4.2.1 Prétraitement des images du corpus

La tâche du prétraitement consiste à appliquer quelques procédures sur les images afin de filtrer le bruit et garder l'information pertinente. Ceci a pour but de rendre efficace et fiable la comparaison entre les différentes images. Le prétraitement consiste essentiellement à l'élimination de bordures d'images tel que montré sur la Figure (4.2). Il est réalisé avec l'algorithme (1) qui se base sur l'initialisation de N germes R_i sur les bords d'images, pour ensuite les propager automatiquement jusqu'à l'étiquetage de tous les pixels de la bande constituant la bordure. Les variables et les fonctions utilisées par l'algorithme(1) sont définies comme suit :

- N est le nombre des germes initialisés automatiquement sur la bordure comme le montre la Figure (4.2), où les germes sont présentés par les points rouges ;
- $\sigma(R_i \cup v)$ est la variance statistique de la région R_i après son augmentation par le pixel voisin v ;
- $C(v)$ est une valeur booléenne (0 ou 1). Cette dernière permet d'indiquer si le voisin v est déjà classé ou pas ;
- δ : seuil de variance tolérée pour la bande.

Ainsi, si la variance $\sigma(R_i \cup v)$ est inférieure au seuil δ et $C(v) = 0$ (n'est pas encore étiquetée), le pixel voisin v sera classé comme une partie de la bordure à éliminer. Ce processus est répété jusqu'à épuisement de tous les pixels de la bordure. Dans l'exemple de la Figure (4.2), la partie (a) montre l'image originale et la partie (b) montre la partie pertinente détectée par l'algorithme(1).

Algorithm 1 Algorithme d'élimination de bordures d'images.

Input : Image avec bordure

Output : Image sans bordure

Soit N germes R_1, \dots, R_N

Initialiser R_1, \dots, R_N

```

1: for  $i = 1 \rightarrow N$  do
2:    $Bool = 1$ ;
3:   repeat
4:     Soit  $V$  l'ensemble des pixels voisins de  $R_i$ ;
5:     if  $\exists v \in V | (\sigma(R_i \cup v) < \delta \text{ et } C(v) = 0)$  then
6:        $R_i \leftarrow R_i \cup v$ ;
7:     else
8:        $Bool \leftarrow 0$ ;
9:     end if
10:  until ( $Bool = 0$ )
11: end for

```

Cette étape de prétraitement facilitera et rendra plus fiable la prochaine étape de notre méthode qui consiste à détecter les différents groupes de lignées en utilisant la partie de l'image qui contient les données utiles (voir la partie cadrée en rouge dans l'exemple).

4.2.2 Identification des groupes de lignées d'images

Dès que la phase de prétraitement est terminée, nous aurons besoin d'identifier les différentes classes ou lignées d'images. On peut définir une lignée comme un groupe d'images qui contient les différentes variantes ou copies de la même image racine ou originale, et qui sont censées appartenir à la même branche de notre graphe final. Pour classer les images en plusieurs groupes de lignées, nous avons combiné deux techniques. La première consiste à identifier les relations de précedence fortes entre

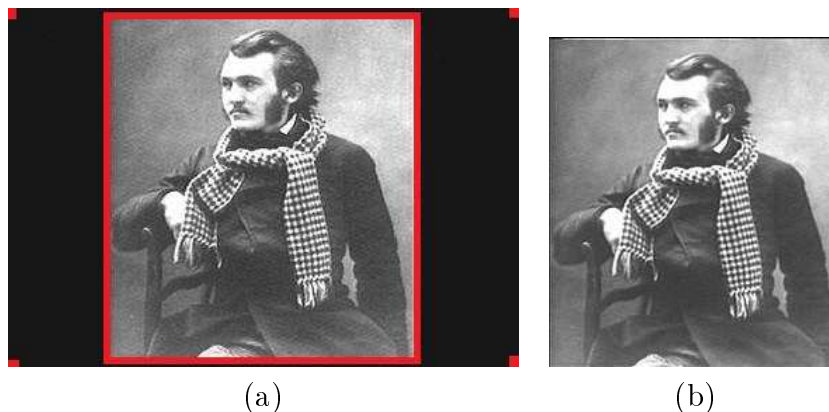


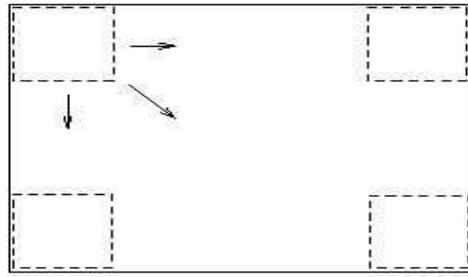
FIGURE 4.2 – Exemple d'un processus de prétraitement

les différentes images et la deuxième sert à classer le reste des images par similitude entre leurs histogrammes.

4.2.2.1 Identification des relations fortes

Des relations fortes peuvent être déterminées entre deux images par de simples calculs sur ces dernières. Par exemple, on peut déterminer si une petite image est une partie d'une autre image plus grande (i.e., détection de recadrage) ou bien si une image est une version en niveau de gris d'une autre image couleur. Les images impliquées dans une relation forte doivent nécessairement appartenir à la même lignée avec une relation de précedence évidente. Dans la suite, nous allons développer les techniques de détection de recadrage et la transformation de couleur en niveau de gris.

a) Détection de recadrage : Nous avons utilisé une technique de détection de recadrage (3.2.5) pour identifier si deux images ou plus peuvent appartenir au même groupe. Si l'on détecte qu'une image est recadrée à partir d'une autre, on peut conclure que ces deux images appartiennent au même groupe. De plus, le processus de détection de recadrage donne des résultats utiles à l'étape suivante, qui est la détection de relation de précedence. Logiquement, si une image A est recadrée à

FIGURE 4.3 – Position (u, v) de l'image modèle sur l'image cible

partir d'une image B , alors nécessairement B précède A .

La technique se base sur l'utilisation de la correspondance d'images (*template matching*). C'est un processus qui consiste à trouver l'emplacement d'une sous-image, appelée un modèle, à l'intérieur d'une autre image cible plus large que le modèle. Cette technique est utilisée dans plusieurs applications en imagerie (ex. recherche d'objets ou détection de formes). Sa mise en œuvre implique souvent *la corrélation croisée normalisée* (NCC), en raison de sa robustesse, pour mesurer la similarité [70, 71]. Pour détecter le recadrage, nous avons proposé un algorithme basé sur le calcul du NCC entre les différentes images. Rappelons que la NCC est une opération utilisée pour mesurer le degré de similitude entre deux signaux. Dans le domaine de l'imagerie, la corrélation peut être utilisée pour calculer le degré de similitude entre deux images ou pour détecter l'emplacement d'une image dans une autre, ce qui nous concerne dans notre travail de détection de recadrage, comme les montre la Figure (4.3).

La valeur du coefficient de la corrélation croisée normalisée entre une image cible X et un modèle (template) T pour chaque point (u, v) , qui a été déplacé par u pixels dans la direction de la coordonnée x et v pixels dans la direction de la coordonnée

y , comme le montre la Figure (4.3), est calculée par l'équation(4.1)

$$C(u, v) = \frac{\sum_{i=-h}^h \sum_{j=-w}^w X(i, j)T(i, j)}{\sqrt{\sum_{i=-h}^h \sum_{j=-w}^w X(i, j)^2} \sqrt{\sum_{i=-h}^h \sum_{j=-w}^w T(i, j)^2}} \quad (4.1)$$

Avec $X(i, j) = X(u + i, v + j)$ et $T(i, j) = T(u + i, v + j)$, où X présente l'image cible, T l'image modèle de hauteur $2h + 1$ et de largeur $2w + 1$. La corrélation croisée normalisée est une mesure invariante aux variations linéaires de la luminosité et du contraste. Elle prend ses valeurs dans l'intervalle $[0, 1]$, où 0 signifie aucune similitude et 1 une similitude parfaite. La Figure (4.4) montre un exemple d'utilisation de corrélation pour la détection de recadrage. Notre algorithme permet de parcourir toutes les images de notre corpus et de détecter si une ou plusieurs images sont recadrées à partir d'une autre. Ceci donne des résultats probants, même si l'image recadrée a subi un changement d'échelle. Comme le montre la Figure (4.4), le visage (b) est recadré à partir de l'image originale (a), puis elle est zoomée. Notre algorithme a réussi à détecter le recadrage, la valeur maximale ou pic de la corrélation entourée par un petit rectangle (c), et la zone de recadrage (d).

Le diagramme (4.5) montre les résultats de détection de recadrage obtenus lorsqu'on applique notre algorithme sur le corpus étudié. Par exemple, l'image 15 est recadrée à partir de l'image 8 et l'image 4 est recadrée à partir de l'image 50, ainsi de suite. Puisque la relation de recadrage est une relation forte, alors ces images doivent appartenir à la même lignée. De plus, la relation de précédence est bien identifiée : par exemple l'image 8 précède automatiquement l'image 15. La Figure (4.6) montre les résultats présentés dans la Figure (4.5) sous la forme d'un graphe généré par la bibliothèque D3.

b) Conversion RGB vers niveau de gris : Ce type de transformation d'une image couleur à une image niveau de gris est expliqué dans la section 3.2.4. On peut considérer cette transformation comme une relation de précédence forte. Autrement

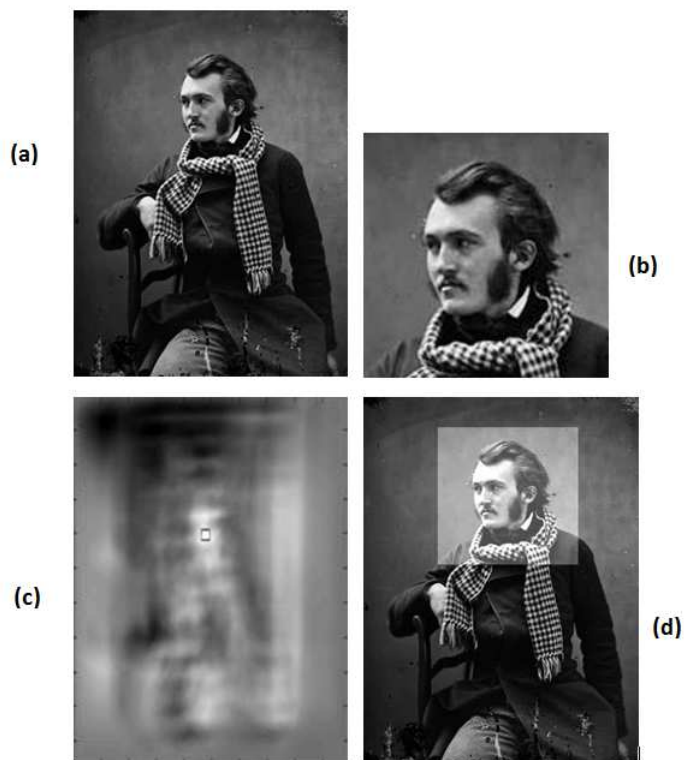


FIGURE 4.4 – Détection de recadrage par CCN

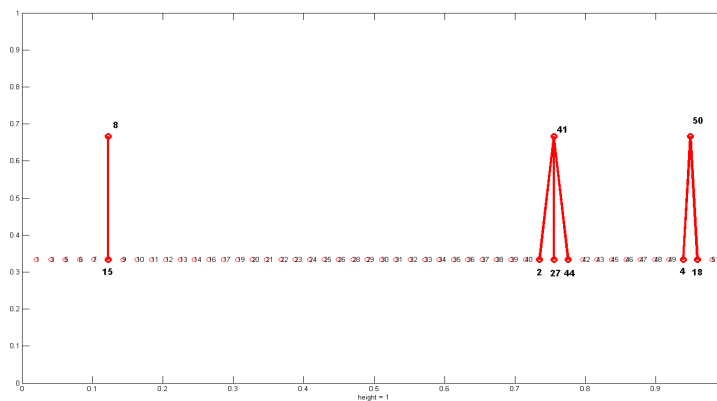


FIGURE 4.5 – Résultats de détection de recadrage



FIGURE 4.6 – Visualisation des résultats avec D3

dit, si une image A est la dérivée d'une image B en couleur, avec A est la version en niveau de gris de cette dernière, alors l'image B précède nécessairement l'image A dans le graphe construit.

4.2.2.2 Classification hiérarchique par distance entre histogrammes

Pour détecter les relations de précédence entre le reste des images, nous avons classé ces dernières par similitude entre leurs histogrammes. Ce processus consiste à classer les images de manière hiérarchique selon la similarité entre les histogrammes de ces dernières par le calcul des distances euclidiennes entre les divers histogrammes. Plus les histogrammes sont proches et similaires, plus la distance entre eux est minimale. Si la distance entre deux histogrammes est égale à 0, les deux images sont supposées être identiques.

La classification hiérarchique ascendante organise les observations définies par un certain nombre de variables, elles-mêmes divisées en modalités, en les regroupant de façon hiérarchique. Elle commence par agréger celles qui sont les plus semblables entre elles, puis on agrège les observations ou groupes d'observations un peu moins semblables et ainsi de suite jusqu'à l'obtention d'un seul groupe. Ces agrégations se font deux à deux à la fois. L'algorithme de classification hiérarchique est composé de deux étapes.

– **Première étape : initialisation de l'algorithme**

- Les groupes initiaux qui sont les n individus de départ. Dans notre cas, ces individus sont présentés par l'ensemble des images constituant notre corpus.
- Calcul de la matrice des distances entre individus deux à deux. La distance utilisée dans notre algorithme est la distance euclidienne entre les histogrammes de différentes images. Soient $H1$ et $H2$ deux histogrammes qui représentent deux images couleur. La distance euclidienne entre $H1$ et $H2$ est définie par :

$$d(H1, H2) = \sqrt{\sum_{j=1}^L (r_{1j} - r_{2j})^2 + \sum_{j=1}^L (g_{1j} - g_{2j})^2 + \sum_{j=1}^L (b_{1j} - b_{2j})^2}, \quad (4.2)$$

où L représente le nombre de niveaux de gris, et r_j , g_j et b_j représentent le nombre de pixels ayant la j ème valeur des canaux de couleur rouge, vert et bleu, respectivement.

– **Deuxième étape : itération de regroupement**

Ici on doit regrouper les deux éléments (individus ou groupes) les plus similaires et mettre à jour le tableau des distances en remplaçant les deux éléments regroupés par le nouveau et en recalculant sa distance avec les autres groupes. Pour la visualisation de cette méthode de classification hiérarchique, nous avons utilisé un arbre hiérarchique aussi appelé *dendrogramme*, présenté dans la Figure (4.7). Ce dernier ne permet pas seulement de montrer les liaisons

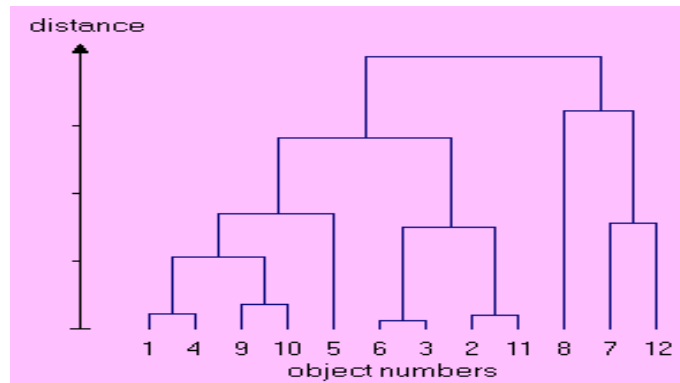


FIGURE 4.7 – Exemple d’un dendrogramme

entre les groupes (l’axe des abscisses), mais la hauteur des branches nous indique le niveau de similarité entre les différents éléments (l’axe des ordonnées).

Pour générer le regroupement hiérarchique de notre corpus d’images, nous avons utilisé la fonction de Matlab $DH = linkage(\mathcal{X}, \text{méthode}, \text{métrique})$, qui permet de créer le dendrogramme DH à partir des données d’images \mathcal{X} en utilisant une méthode spécifiée *méthode*, décrivant la façon de mesurer la distance entre les groupes, et la *métrique* décrivant le type de distance utilisée (ex. Euclidienne, etc). Nous avons utilisé la fonction avec les paramètres suivants :

$$DH = linkage(\mathcal{X}, \text{Single}, \text{Euclidean});$$

Avec :

- \mathcal{X} : représente les histogrammes des différentes images ;
- *Single(voisin le plus proche)* : c’est l’une des techniques hiérarchiques les plus anciennes et les plus connues. La distance entre les deux groupes est définie comme étant la distance entre les deux histogrammes les plus proches entre les deux groupes ;

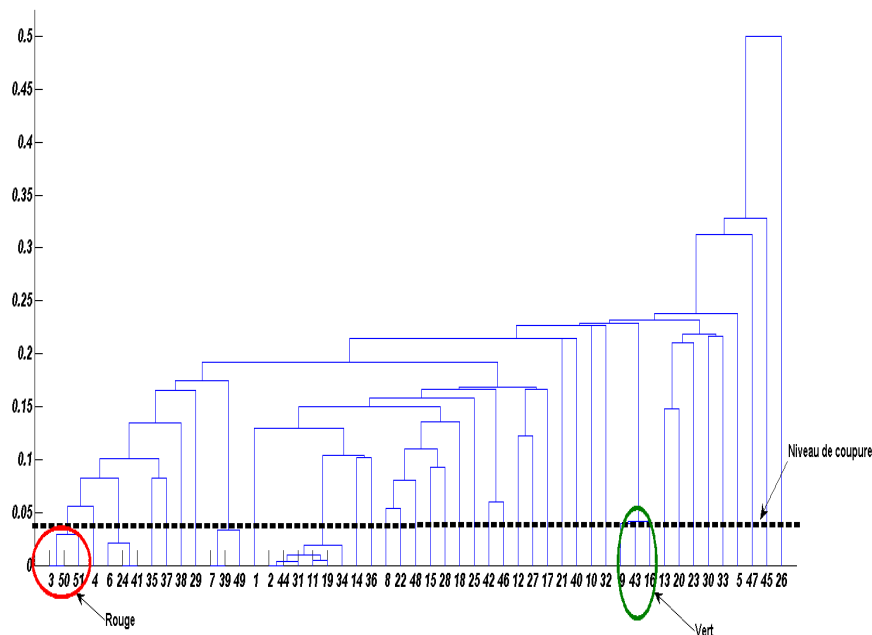


FIGURE 4.8 – Classification du corpus étudié par distance entre histogrammes

- *Euclidean* : métrique de calcul de distance entre les différents histogrammes. Le calcul de cette distance est présenté par l'équation (4.2).

Nous avons appliqué cet algorithme de classification hiérarchique sur notre corpus d'images. La Figure (4.8) montre les résultats obtenus pour notre corpus, avec l'axe des abscisses représentant les numéros des images et l'axe des ordonnées représentant le niveau de similitude ou la distance entre les histogrammes des différentes images constituant notre corpus.

Pour que le résultat soit plus clair, un échantillon de notre dendrogramme est montré sur la Figure 4.9 pour le groupe contenant les images 3, 51, 50 et 4 (encadrés en rouge) et les images 9, 43 et 16 (encadrés en vert). Comme le montre le dendrogramme (4.8), la distance entre les images 3 et 50 est égale à 0, ce qui signifie que les deux images sont similaires. L'image qui a une distance minimale avec cette agrégation est la 51, puis la 4, et ainsi de suite. Un autre

regroupement présenté à la droite de la Figure (4.9) présente les résultats du regroupement encerclé en vert et qui montre aussi une grande ressemblance entre les images regroupées.

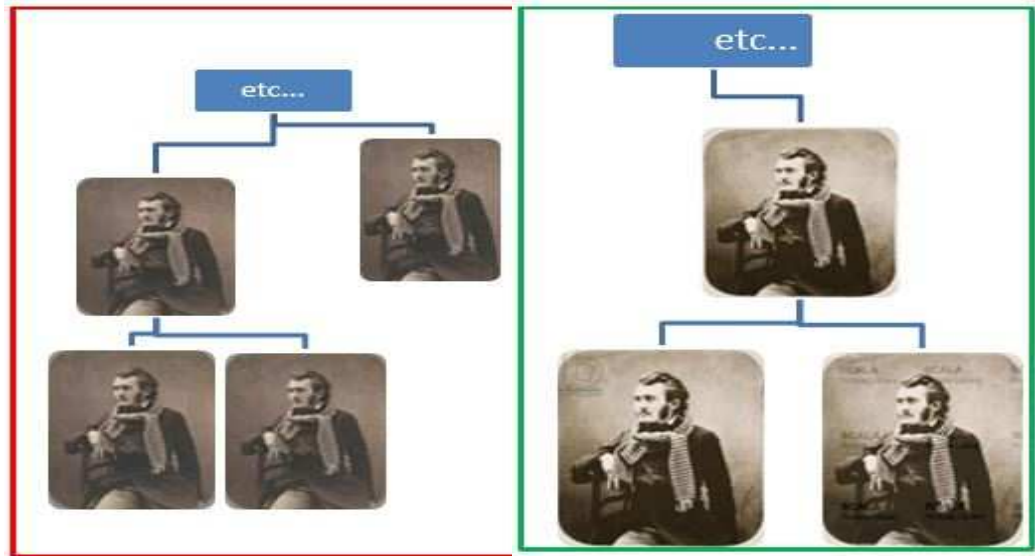


FIGURE 4.9 – Exemple de regroupement d'images par distance entre histogrammes

– **Troisième étape : identification finale des groupes**

On suppose que l'on dispose d'un nombre d'images originales que l'on peut considérer comme des *images racines* dans le graphe. Une image racine est une image identifiée comme telle par un expert, et qu'on peut utiliser physiquement comme une racine d'une lignée dans le graphe. Nous avons commencé par n groupes où chacun contient une seule image racine. Puis, nous avons fusionné les groupes d'une manière hiérarchique en appliquant l'algorithme de classification hiérarchique expliqué dans la section (4.2.2.2), jusqu'à ce que l'on obtienne deux images originales dans le même groupe. Ce processus est expliqué par l'algorithme (2) qui renvoie la seuil c au delà duquel les fusions de groupes ne seront pas permises.

Dès qu'on obtient la valeur du seuil c par l'algorithme, on applique la fonction de Matlab : $G = Cluster(DH, Cutoff, c)$. Cette fonction permet de construire des groupes G à partir d'une classification hiérarchique, avec DH le résultat généré par

Algorithm 2 Identification finale des groupes

Input : n images**Output** : les lignées constituant le corpus étudié**Soit n groupes** G_1, \dots, G_n

(4.2.2.2)

```

1:  $Fusion = 1$ ;
2: repeat
3:   Soit  $G_x, G_y$  les deux prochains groupes à fusionner ;
4:    $O_x \leftarrow$  Nombre d'images racines dans le groupe  $G_x$ ;
5:    $O_y \leftarrow$  Nombre d'images racines dans le groupe  $G_y$ ;
6:   if  $O_x + O_y < 2$  then
7:     fusionner les deux groupes  $G_x$  et  $G_y$ ;
8:     recalculer les distances entre les groupes
9:   else
10:     $c \leftarrow$  distance entre  $G_x$  et  $G_y$ ;
11:     $Fusion \leftarrow 0$ ;
12:   end if
13: until ( $Fusion = 0$ )
14: fusionner les groupes sans racine avec les groupes avec racine les plus proches ;

```

la fonction *linkage* présentée dans la section (4.2.2.2) et c le seuil de coupure. L'option *Cutoff* permet de couper le dendrogramme dans le niveau de similitude c .

Les groupes sont formés quand la valeur de la distance entre un groupe et tous ses sous-groupes est inférieure à la valeur de coupure. Toutes les feuilles situées dans le même niveau ou positionnées en-dessous d'un groupe sont fusionnées pour former un seul groupe. Dans notre corpus, les deux premières images racines fusionnées sont 9 et 43, avec une distance entre les deux groupes égale à 0.394. La coupure de dendrogramme avec cette valeur permet d'obtenir 37 groupes ou lignées, comme le montre la Figure (4.8). Pour le reste des images situées en-dessus du niveau de coupure nous avons fusionné chaque image avec le groupe qui contient l'image racine la plus proche à cette dernière.

4.2.2.3 Construction et visualisation des relations de précedence

Après l'identification des différentes lignées ou groupes, l'étape suivante de notre méthode consiste à détecter les relations entre les différents éléments de chaque groupe dans le but d'identifier les relations de précedence entre ces derniers. Pour atteindre cet objectif, nous avons implémenté un algorithme qui permet de parcourir les éléments de chaque groupe et de les comparer deux à deux avec l'analyse de corrélation entre les images.

Comme vu précédemment, la corrélation permet de calculer le degré de similarité entre deux images. Dans la section précédente, nous avons utilisé la CCN pour détecter si une image est recadrée à partir d'une autre image. Dans le but de calculer le score entre les différentes images, nous avons utilisé la corrélation pour avoir un score de similarité entre les diverses images de chaque groupe. La valeur de la corrélation est comprise entre 0 et 1. Plus les images sont similaires, plus cette valeur est proche de 1. Si la corrélation entre deux images est proche de 1 alors ces deux dernières sont fortement similaires.

Après le calcul des valeurs de corrélation entre les différents éléments de chaque lignée, elles sont utilisées par l'algorithme (3) pour construire les relations de précedence entre les images. Les variables de l'algorithme sont définies comme suit :

- Score : est la valeur de corrélation entre deux images I1 et I2.
- G_v : représente le groupe courant.
- G_c : représente l'ensemble d'images connectées au graphe.
- \bar{G}_c : représente l'ensemble d'images non connectées au graphe ($\bar{G}_c \leftarrow G_v - G_c$).

L'algorithme (3) proposé est constitué des étapes suivantes :

1. Initialiser l'ensemble G_c par l'image racine du groupe G_v .

2. Parcourir l'ensemble \bar{G}_c et G_c dans le but de chercher ind_1 et ind_2 , c'est à dire les indices de deux images, ayant le score maximum de l'ensemble G_c et \bar{G}_c respectivement.
3. Ajouter l'image non connectée $\bar{G}_c[ind_2]$ qui possède le score maximum avec l'image $G_c[ind_1]$ et tous ses descendants à l'ensemble des éléments connectés G_c comme des fils de l'élément connecté $G_c[ind_1]$.
4. Supprimer l'élément $\bar{G}_c[ind_2]$ et tous ses descendants à partir de l'ensemble d'éléments non connectés \bar{G}_c .
5. Répéter les étapes (2, 3 et 4) jusqu'à ce que le nombre des éléments de l'ensemble connecté G_c soit égal au nombre des éléments de groupe G_v (c'est-à-dire que tous les éléments de groupe G_v seront connectés au graphe).

Nous avons appliqué cet algorithme de détection de précédence sur les différentes lignées. La Figure (4.10) montre les résultats obtenus.

Suite à l'identification des groupes et la construction des relations de précédence entre les éléments de chaque groupe, la Figure (4.11) présente le graphe de résultat. Il montre les différentes lignées, chacune contenant une image racine et ses dérivées de manière hiérarchique. Cette figure illustre les résultats sous la forme d'un arbre généré par Matlab. La Figure (4.12) montre une lignée extraite à partir du graphe présenté ci-dessus, générée par la bibliothèque D3. Comme on voit sur la figure, les relations de précédence sont bien identifiées (par exemple l'image 41 est une image racine qui précède les images 2, 27, 24 et 44) et ainsi de suite.

4.2.3 Analyse des éditions d'images

Après la construction des différentes lignées et relations de précédence entre les images du corpus, nous allons tenter à identifier les différentes transformations effec-

Algorithm 3 Construction des relations de précedence**Input** : les lignées constituant le corpus étudié**Output** : un graphe hiérarchique : identifier les relations de précedence entre les différents éléments de chaque lignée $K \leftarrow$ nombre de groupes ;

```

1: for  $v = 1 \rightarrow K$  do
2:   soit  $r_v$  la racine de groupe  $G_v$  ;
3:    $G_c \leftarrow r_v$  ;
4:    $\bar{G}_c \leftarrow G_v - G_c$  ;
5:    $m \leftarrow$  nombre d'images dans le groupe  $G_v$  ;
6:   if  $m > 1$  then
7:     for  $i = 1 \rightarrow m$  do
8:       for  $j = 1 \rightarrow m$  do
9:         if  $\text{recadrage}(\text{image}[i]) = \text{image}[j]$  then
10:          image[i] précède image[j] ;
11:         end if
12:         if  $\text{Score}(\text{image}[i], \text{image}[j]) = 2$  then
13:          les deux images sont similaires ;
14:         end if
15:       end for
16:     end for
17:     while  $|G_c| < m$  do
18:       initialiser une matrice nulle M de taille  $(|G_c|, |\bar{G}_c|)$  ;
19:       for  $j = 1 \rightarrow |G_c|$  do
20:         for  $k = 1 \rightarrow |\bar{G}_c|$  do
21:            $M(j,k) \leftarrow \text{Score}(G_c(j), \bar{G}_c(k))$  ;
22:         end for
23:       end for
24:        $[ind_1, ind_2] \leftarrow \text{arg max}_{(j,k)}(M)$  ;
25:        $G_c[ind_1]$  précède  $\bar{G}_c[ind_2]$  ;
26:        $D \leftarrow$  les descendants de  $\bar{G}_c[ind_2]$ 
27:        $G_c \leftarrow G_c \cup \bar{G}_c[ind_2] \cup D$  ;
28:       supprimer  $\bar{G}_c[ind_2]$  et  $D$  à partir de  $\bar{G}_c$  ;
29:     end while
30:   end if
31: end for

```

tuées entre les images. En particulier, nous identifierons les types d'édicions effectuées sur les images depuis leur création. Pour ce faire, nous utilisons l'apprentissage automatique supervisé pour détecter la présence d'un traitement d'édition entre chaque image et ses dérivées, ainsi qu'identifier les types d'édicions.

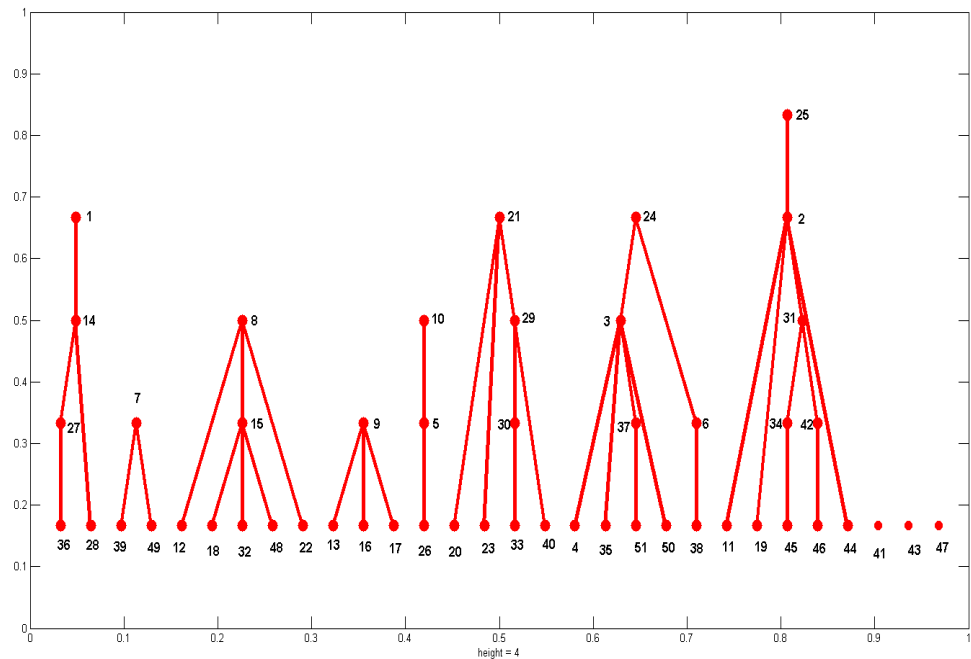


FIGURE 4.10 – Résultats de construction de relations de précedence

L'apprentissage automatique (*machine-learning*) est l'un des champs d'étude de l'intelligence artificielle. Ce dernier fait référence à la capacité d'un système à acquérir et intégrer de façon autonome des connaissances pour une meilleure prédiction [19]. L'un des domaine d'apprentissage est *la classification supervisée* qui consiste à apprendre un modèle de classification à partir de données d'apprentissage, pour pouvoir par la suite prédire la classe de nouvelles données. Dans l'ensemble d'apprentissage, chaque exemple est constitué d'une description sous forme de vecteur et d'une étiquette désignant la classe.

En ce qui concerne les algorithmes d'apprentissage supervisé, on fournit à l'algorithme des données d'entraînement $D_n = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$ de n exemples où chaque exemple est une paire $(x^{(i)}, y^{(i)})$, $x^{(i)}$ constitue les données d'entrée et $y^{(i)}$ la cible. Chaque instance de données possède D entrées qui sont les attributs (caractéristiques) des données. On dénote par \mathcal{X} l'ensemble des entrées pos-

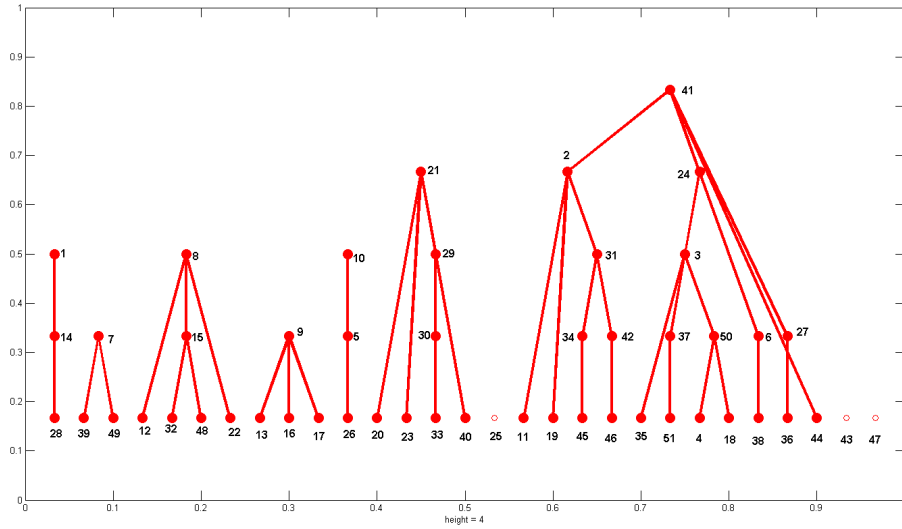


FIGURE 4.11 – Résultat final généré par Matlab

sibles et par \mathcal{Y} l'ensemble des sorties possibles. Souvent $\mathcal{X} = \mathbb{R}^D$ et $\mathcal{Y} = \{1, 2, \dots, K\}$ pour une classification à K classes. On dispose donc d'un ensemble initial de données dont la classification est connue. Ces données sont supposées être indépendantes et identiquement distribuées, et serviront à l'apprentissage d'un modèle de classification qui sera utilisé pour prédire la classe de nouvelles données.

4.2.3.1 Détection de la présence d'édition

Pour détecter la présence d'une édition entre une image originale et sa dérivée, nous avons utilisé la mesure de qualité d'image SSIM (indice de similitude structurale) [72] entre ces images. Cette valeur représente une caractéristique mesurant le degré de dégradation d'une image par rapport à une image référence (idéale ou parfaite). La mesure SSIM entre deux images x et y est calculée par l'équation suivante :

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)(2cov_{xy} + c_3)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)(\sigma_x\sigma_y + c_3)} \quad (4.3)$$

Avec :

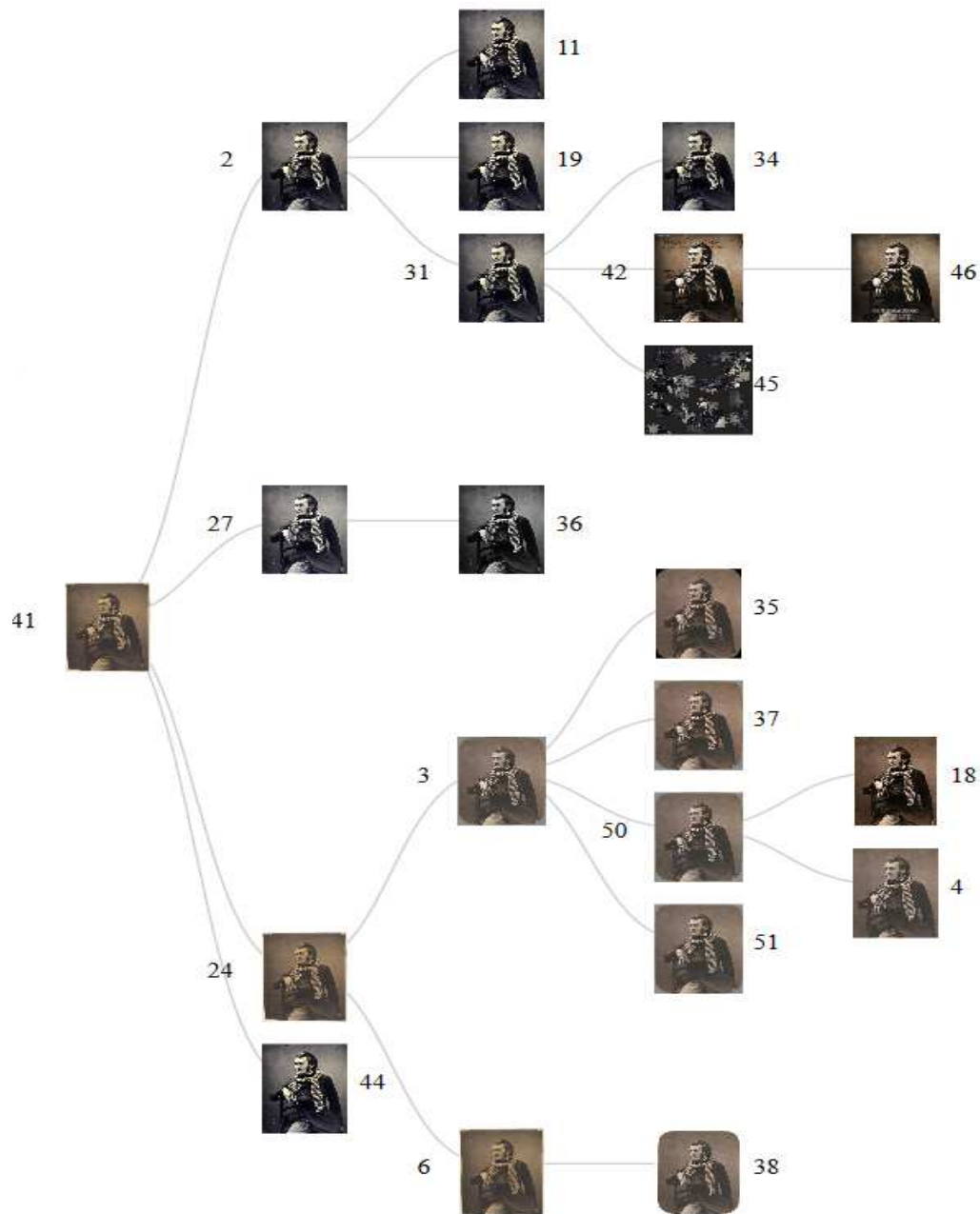


FIGURE 4.12 – Graphe final des relations de précédence généré par D3

1. μ_x la moyenne de x ;
2. μ_y la moyenne de y ;
3. σ_x^2 la variance de x ;
4. σ_y^2 la variance de y ;
5. cov_{xy} la covariance de x et y ;



FIGURE 4.13 – Exemple des différentes valeurs de SSIM

Dans le contexte de notre étude, nous avons utilisé la mesure de qualité d'image SSIM pour calculer un score de dégradation entre les images d'une lignée. L'objectif de SSIM est de mesurer la similarité de structure entre deux images, en supposant que l'image référence est de qualité supérieure. La valeur de la fonction SSIM est comprise entre 0 et 1. Plus cette valeur est proche de 1, plus la qualité de l'image dégradée est bonne et proche de l'image originale. La Figure (4.13) montre une mesure de SSIM sur différentes images dégradées à partir de la même image originale.

Notons que pour les images qui ont une valeur de SSIM forte, nous avons utilisé aussi l'indice de corrélation CCN pour vérifier si l'image a subi seulement une transformation d'illumination. Les valeurs de CCN sont calculées entre l'image originale et ses dérivées après l'égalisation des histogrammes sur ces deux dernières (section(3.2.1)). Si cette valeur est égale à 1, alors l'image dérivée a subi véritablement un changement d'illumination, et auquel cas aucune analyse supplémentaire d'édition n'est requise. Nous avons appliqué cet algorithme sur notre corpus. La Figure (4.14) montre un échantillon des résultats obtenus, où les images 11, 19 et 31 sont des copies de l'image 02, qui ont subi une transformation d'illumination, et

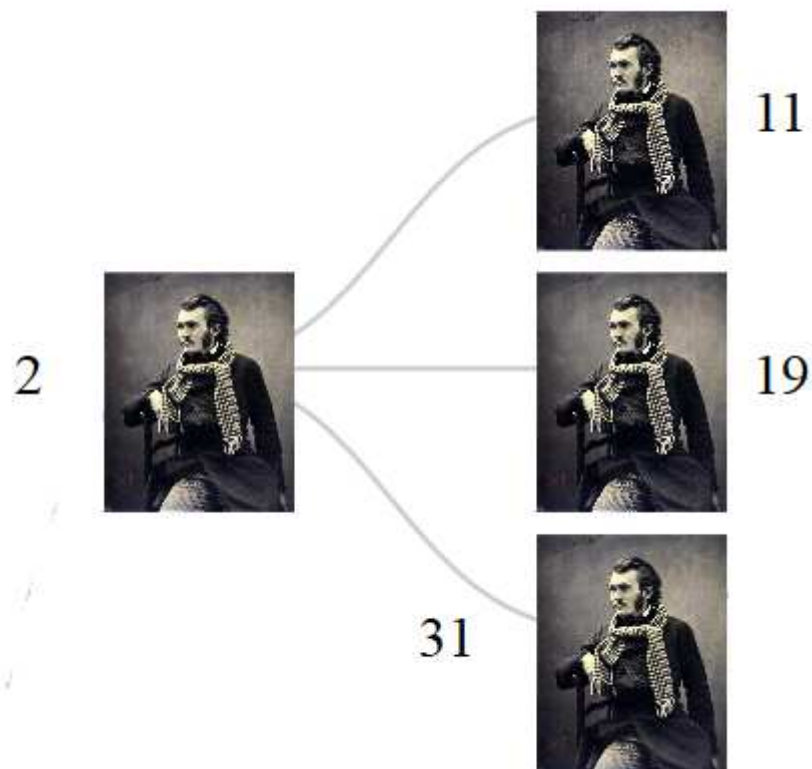


FIGURE 4.14 – Exemple de trois variantes qui ont subi un processus d'illumination elles ont été détectées en tant que telles par notre méthode.

Dans le but de détecter la nature des transformations appliquées sur les images de notre corpus, nous avons créé un scénario avec 220 images dérivées à partir de la même image originale. Ces dernières ont subi plusieurs types de transformations (niveau de gris, couleurs, insertion de texte, insertion d'objets, etc.). Nous avons calculé les différentes valeurs de SSIM entre les images dérivées et l'image originale après avoir égalisé leurs histogrammes, puis nous avons calculé les fréquences de SSIM présentées sur le graphe de Figure (4.15).

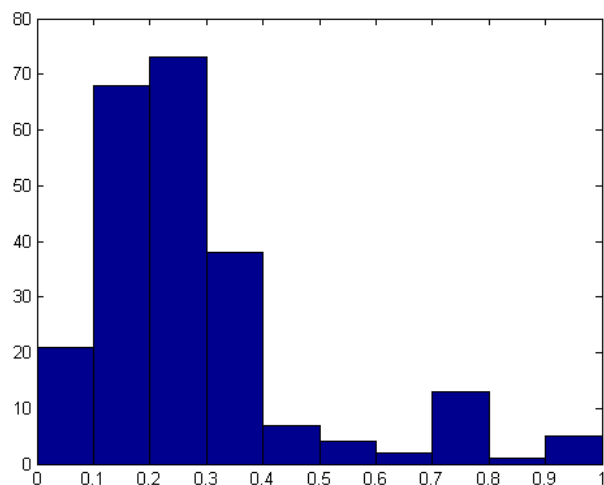


FIGURE 4.15 – Histogramme des valeurs SSIM de modèle étudié

Nous avons observé que les images qui ont une valeur de SSIM supérieure ou égale à 0.6 ($SSIM > 0.6$) ont souvent subi des modifications mineures, autrement dit il n'y a pas d'édition dans l'image (insertion d'objets, de textes ou de cadres dans l'image, etc.). Ces transformations touchent les valeurs de niveaux de gris sur toute l'image (luminosité, contraste, etc.). Par contre, les images qui ont des valeurs de SSIM inférieures à 0.6 ont subi généralement des transformations majeures consistant à appliquer des processus d'édition sur l'image originale (insertion de texte, d'objet, de cadre, etc.). En effet, plus l'image a subi des processus d'édition, plus la valeur SSIM sera proche de 0. La Figure 4.16 montre quelques valeurs de SSIM entre une image originale et trois variantes dérivées à partir de cette dernière.

La valeur de SSIM entre deux images peut être un bon indice de présence/absence d'une édition. En se basant sur les valeurs de SSIM, nous avons utilisé l'apprentissage automatique supervisé basé sur la régression logistique pour calculer la probabilité de présence/absence d'une édition. Nous avons utilisé notre ensemble d'entraînement constitué de $n = 220$ images et présenté dans la section précédente. Chaque image



FIGURE 4.16 – Exemple de l’influence de traitement d’édition sur le SSIM : a) image originale, b) $ssim=0.76$ et c) $ssim=0.54$ et d) $ssim=0.15$

dans cet ensemble est annotée par la présence ($y = 1$) ou l’absence ($y = 0$) d’édition.

La régression logistique est une méthode de classification basée sur la maximisation de probabilité a posteriori des classes par rapport aux données d’apprentissage. On distingue habituellement la régression logistique ordinaire ou binaire de la régression logistique multinomiale. La première vise à expliquer une classification de type binaire (ex. oui/non, vrai/faux, etc.). La deuxième est une extension de la régression logistique binaire pour plusieurs classes ($K > 2$). D’une manière générale, le modèle de régression logistique considère que le logarithme de la vraisemblance des données peut être décrit comme une fonction linéaire du vecteur de caractéristiques des objets à classifier. Pour le cas de 2 classes, cela revient à trouver le vecteur qui définit un hyperplan qui constituera la frontière de décision séparant les deux classes édition/non-édition. Pour estimer cette frontière, on maximise la vraisemblance des classes et données en approximant la probabilité a posteriori des classes par une fonction sigmoïde agissant sur un vecteur de données à D attributs $x = (x_1, x_2, \dots, x_D)$ (dans notre cas $D = 1$), comme suit :

$$P(y = 1|x) = 1/(1 + \exp(-f(x))) \quad (4.4)$$

Où $f(x) = w_0 + \sum_{d=1}^D (w_d x_d)$ est l'équation de l'hyperplan séparant les deux classes (éditer, non-éditer). (w_0, w_1, \dots, w_D) est le vecteur des coefficients de régression qui définit un hyperplan qui sépare les deux classes. $y \in \{0, 1\}$ est la valeur cible qui prend comme valeur 1 lors de la présence d'une d'édition et 0 sinon.

Nous avons appliqué la méthode de la regression logistique pour calculer la probabilité d'édition entre une image et sa dérivée. La Figure(4.17) illustre quelques résultats obtenus entre une image racine(a) et ses dérivées(b, c et d). La Figure(4.18) présente un exemple récapitulatif des résultats obtenus après la construction des relations de précédence et la détection de la présence de différents types de transformation (recadrage, conversion couleur vers niveau de gris, édition et illumination) entre les diverses images. Par exemple il existe un traitement d'édition entre l'images 3 et 35 avec une probabilité égale à 0.7.

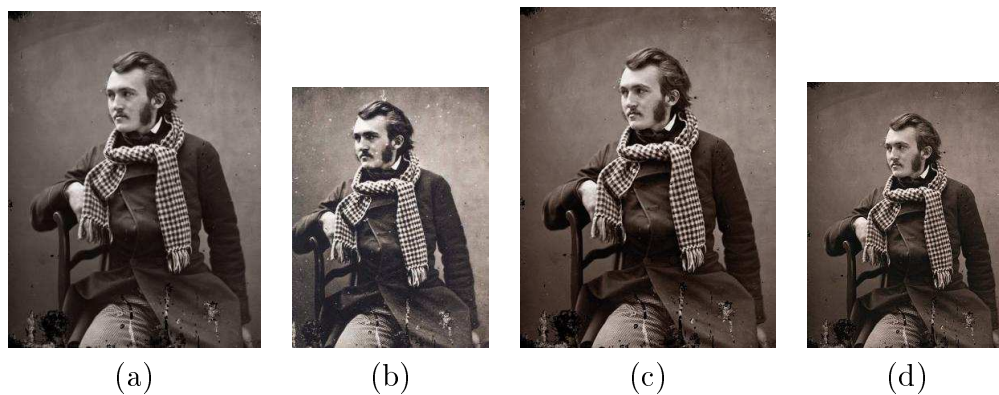


FIGURE 4.17 – Exemple des probabilités d'édition : a) image originale, b) $P(\text{éditer})=0.8$ et c) $P(\text{éditer})=0.2$ et d) $P(\text{éditer})=0.1$

4.2.3.2 Catégorisation probabiliste des éditions

Dans la section précédente, nous avons calculé la probabilité qu'une image dérivée à partir d'une image parent ait subi un traitement d'édition. Dans le but d'identifier le(s) type(s) de transformation(s) effectuée(s) sur une image dérivée, nous explorons une démarche pour classer les transformations d'éditions en catégories. Nous

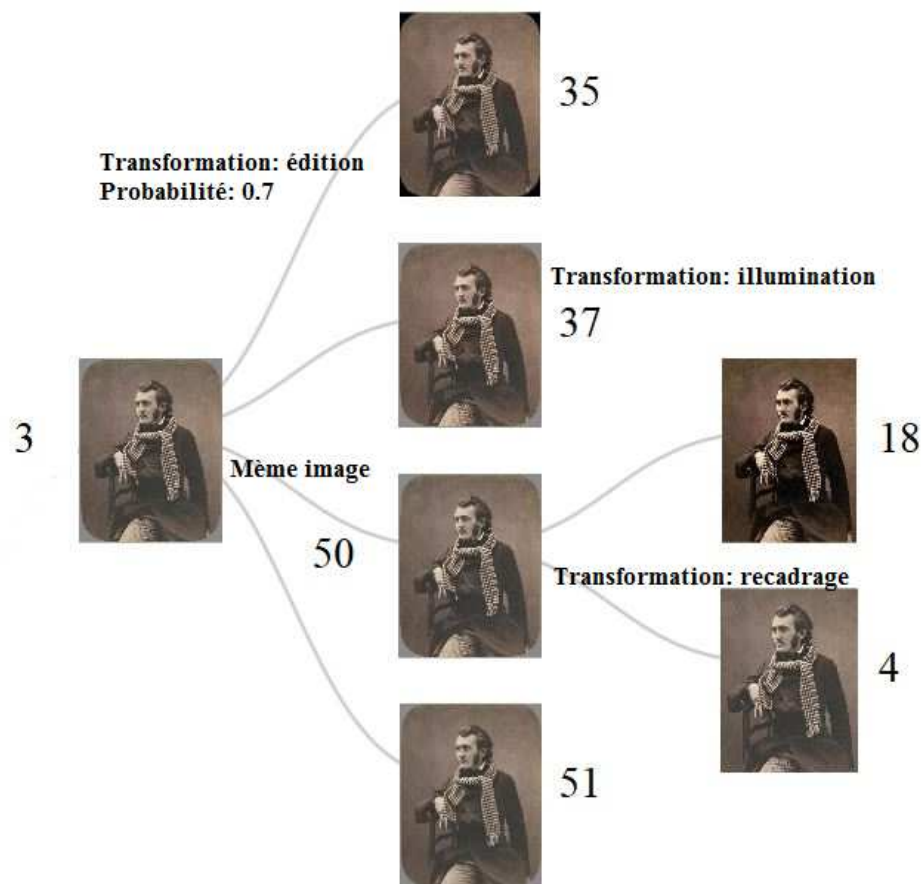


FIGURE 4.18 – Exemple d’un échantillon des différents types de transformation

avons recensé trois types d’édérations qui sont usuellement subies par les images de photographie :

- **Écriture manuelle** : ce type d’édition est illustré par la Figure (4.19(a)). Il est défini comme l’ajout d’une écriture manuelle à une image originale (ajout d’un nom, signature, etc.).
- **Insertion d’objet** : on peut définir cette transformation comme l’insertion d’un objet dans l’image (visage, forme géométrique, etc.) comme le montre la Figure (4.19(b)), où l’image originale a été éditée par l’insertion d’un petit rectangle noir.

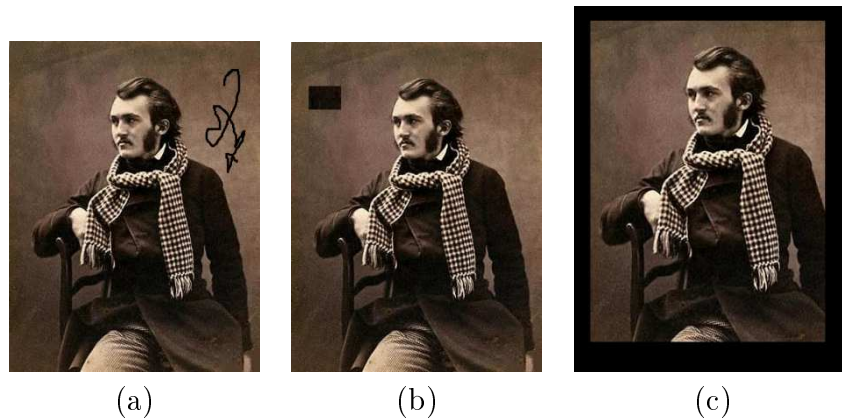


FIGURE 4.19 – Différents types de transformations : a) écriture manuelle, b) insertion d'objet et c) insertion d'un cadre

- **Insertion d'un cadre** : cette transformation consiste à ajouter un cadre à une image, comme montre la Figure (4.19(c)).

Pour extraire les différentes régions éditées entre une image et sa dérivée, nous avons appliqué un traitement de soustraction entre l'image éditée et la version qui la précède si les deux images ont la même taille . Dans le cas échéant, nous avons appliqué un processus d'alignement entre les images en utilisant la détection de la similarité de contenu avec CCN. De cette manière on détecte les régions modifiées entre ces deux dernières. Notons qu'on peut détecter plusieurs régions éditées, mais nous avons éliminé les petites régions et nous avons gardé les régions les plus grandes. En effet on peut détecter des petites régions comme de régions éditées à cause de différentes manipulations et traitements exercés sur l'image dérivée, ces dernières peuvent exercer des changements dans les valeurs des pixels. Pour chaque région d'image où une édition est présente entre une image originale et sa dérivée, les propriétés géométriques suivantes de l'édition ont été calculées :

- *L'aire* : valeur scalaire indiquant le nombre de pixels dans la région éditée.
- *L'excentricité* : valeur scalaire qui mesure l'excentricité de l'ellipse qui a les mêmes seconde-moments que la région éditée.

-
- *Le périmètre* : valeur scalaire qui mesure le nombre de pixels entourant la frontière de la région éditée.
 - *La solidité* : valeur scalaire indiquant la proportion des pixels dans l’enveloppe convexe qui sont également dans la région éditée.

La probabilité d’appartenance de chaque région détectée à l’une des trois catégories d’édition est alors calculée en utilisant les $D = 4$ caractéristiques. Noter qu’une image peut être assignée à plus d’une catégorie d’édition dépendamment du nombre et les types d’édition détectées. Soient C_1 , C_2 et C_3 les trois catégories d’édition respectivement : insertion de texte, inertion d’objet et insertion de cadre. Pour détecter ces différents types de transformation utilisés entre deux images, nous avons utilisé la regression logistique multinomiale sur notre ensemble de données. La probabilité a posteriori de chaque catégorie est définie par :

$$P(y = C_i|x) = \frac{\exp^{-W^{(i)}x}}{\sum_{j=1}^K \exp^{-W^{(j)}x}} \quad (4.5)$$

Où $W^{(i)}$ est un vecteur de poids (ou coefficients de régression) correspondant au catégorie k . $x = (x_1, x_2, \dots, x_D)$ est l’ensemble des variables décrivant chaque observation. Dans notre cas, cet ensemble est constitué de 4 éléments expliqués précédemment (aire, excentricité, périmètre, solidité) et $y \in \{C_1, C_2, C_3\}$ est la valeur cible qui présente la catégorie d’édition.

Nous avons remarqué à partir des valeurs de 4 caractéristiques étudiées que les régions éditées de type insertion d’une écriture manuelle ont une valeur plus petite d’area et sont moins solides par rapport aux éditions de type insertion de cadre et d’objet. Par contre cette dernière possède une valeur d’excentricité élevée par rapport les deux autres catégories d’édition. Dans le cas des des régions éditées de type insertion de cadre et d’objet, ces dernières ayant en général une valeur d’area

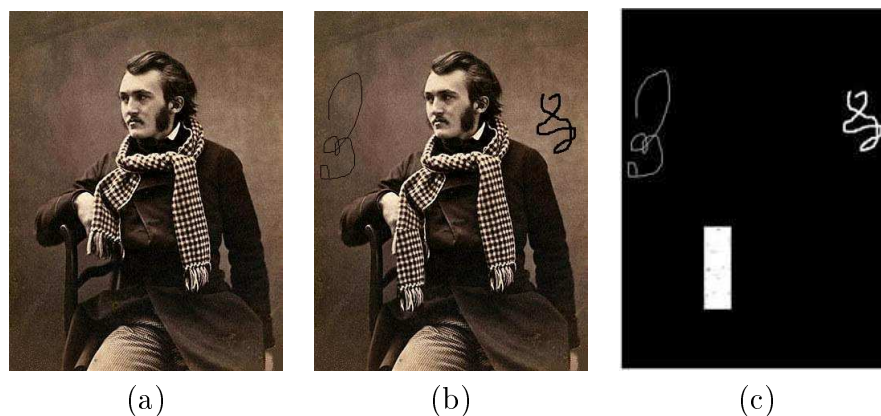


FIGURE 4.20 – Exemple de détection de l'édition : a) image originale, b) image éditée et c) régions détectées

Numéro de la région	C_1	C_2	C_3
1	0.99	$\simeq 0$	$\simeq 0$
2	$\simeq 0$	0.99	$\simeq 0$
3	1	$\simeq 0$	$\simeq 0$

TABLE 4.1 – Probabilité d'appartenance de chaque région à une catégorie d'édition : C_1 (Écriture manuelle), C_2 (Insertion d'objet) et C_3 (Ajout de cadre).

plus grande et sont aussi plus solides et moins excentriques par rapport à la catégorie d'une écriture manuelle.

La Figure 4.20 montre le résultat de classification d'éditions obtenu en appliquant notre algorithme sur une image originale et sa dérivée contenant trois types d'édition.

Le tableau (4.1) illustre les résultats de calcul de la probabilité du chaque type d'édition des régions éditées détectées dans la Figure (4.2.3.2), avec les lignes présentant le numéro d'édition détecté de gauche à droite, et les colonnes présentant la probabilité du chaque type d'édition.

Nous avons appliqué cet algorithme sur chaque couple d'images voisines de notre corpus étudié qui ont subi une transformation d'édition. La figure (4.21) montre un exemple d'une image originale (a) et sa dérivée éditée (b), ainsi que les régions éditées détectées (c).

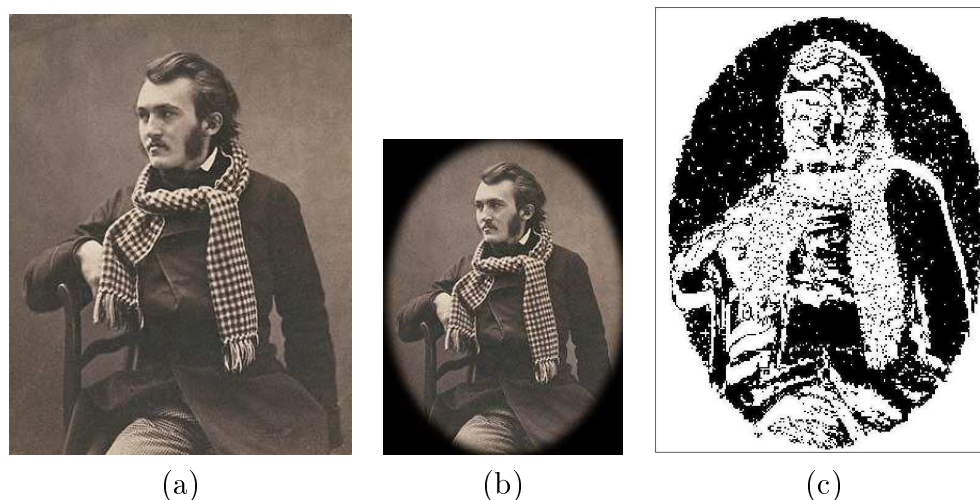


FIGURE 4.21 – Exemple de détection de l'édition : a) image originale, b) image éditée et c) régions détectées

Le tableau (4.2) montre les résultats de catégorisation de la région éditée détectée dans la figure (4.21). Comme on voit sur cette dernière, les deux images ne sont pas de la même taille et notre algorithme a réussi à détecter la présence d'une région éditée avec une probabilité égale à (0.9) que cette dernière est de type insertion de cadre.

	C_1	C_2	C_3
région éditée	0.02	$\simeq 0.08$	$\simeq 0.9$

TABLE 4.2 – Probabilité d'appartenance de la région éditée à une catégorie d'édition : C_1 (Écriture manuelle), C_2 (Insertion d'objet) et C_3 (Ajout de cadre).

4.2.4 Intégration de métadonnées dans la classification

Nous avons décrit dans la section(3.4.1) les différents types de métadonnées d'images. L'analyse des métadonnées des fichiers numériques peut en effet contribuer à mieux comprendre les changements opérés sur une image et les dates de leurs applications, et ainsi suivre l'évolution temporelle des images constituant notre corpus. Cependant, il existe plusieurs limitations dues principalement aux logiciels manipulant les images et les systèmes de gestion d'images sur le Web. En effet, l'inexactitude

des métadonnées et la perte d'information liée à la circulation en ligne est un problème commun pour les images circulant sur le web.

Lorsque les images sont téléchargées sur les plateformes web, une grande partie de leurs métadonnées est supprimée, en particulier dans les versions les plus petites et les formats miniatures générés par des applications Web telles que les réseaux sociaux (Facebook, Twitter, WordPress, etc.). Le respect de la vie privée peut être l'une des raisons pour lesquelles cette information est supprimée, mais le principal objectif est d'alléger les fichiers et de faciliter le stockage et l'affichage des images en ligne [22]. C'est le cas de notre corpus où la plupart des images ne possèdent pas de métadonnées en raison des limites mentionnées dans la section (3.4.4). De plus, même en cas de présence de ces métadonnées, les risques d'erreur sont particulièrement élevés en ce qui concerne les dates à cause des raisons suivantes [22] :

- L'appareil photo ne tient pas compte de son fuseau horaire local, ce qui provoque des décalages dans les dates de création des images [17].
- Il existe toujours un risque d'inexactitude des informations enregistrées par l'ordinateur dans le cas des métadonnées. En effet, il suffit de déplacer le dossier où l'image est enregistrée ou même d'ouvrir le fichier puis de l'enregistrer sans modifier l'image pour changer les dates de création ou de modification situées dans les métadonnées [52].

Ces obstacles peuvent conduire à une mauvaise interprétation des différentes informations de métadonnées lors de la détection des relations de précedence. En raison de ces différents inconvénients et de l'absence de métadonnées dans la plupart des images de notre corpus, l'intégration des métadonnées dans notre classification est non fiable et donc laissée ouverte pour de futures possibilités.

4.3 Sommaire du chapitre

Dans ce chapitre, nous avons proposé une méthode qui permet, dans un premier lieu, la détection des différentes lignées constituant notre corpus. Ceci est effectué par la découverte des différentes relations de précédence entre les éléments de chaque lignée. Dans un deuxième temps, nous avons utilisé l'apprentissage automatique supervisé pour calculer la probabilité qu'une image dérivée ait subi une transformation d'édition et nous avons catégorisé les éditions en identifiant la nature des transformations entre chaque couple d'image ayant une relation de précédence.

Chapitre 5

Conclusion générale

L'analyse des collections d'images est devenue un outil indispensable pour comprendre les échanges de données sur les réseaux sociaux et la communication sur le Web. Un des problèmes de la gestion des collections d'images est la délimitation des corpus d'images reliées à une requête et la visualisation de ces dernières pour faciliter l'interprétation et la compréhension des relations entre les images. Une des informations utiles recherchées est la visualisation de l'ordre chronologique des copies d'une même image originale. Au meilleur de nos connaissances, aucune méthode n'a été proposée pour ce problème.

Ce mémoire vient combler ce vide en proposant une méthodologie novatrice pour la visualisation de l'ordre temporel de copies d'images. Après une présentation d'une revue de littérature sur les méthodes de recherche et de visualisation des grandes collections d'images, nous avons exposé les différentes altérations qu'une image peut subir avec une description mathématique des processus physiques. Nous avons par la suite exposé les détails de notre méthodologie visant à visualiser l'évolution temporelle d'un corpus d'images contenant différentes copies/variantes en circulation sur le Web de la même image.

Pour estimer les relations de précedence entre les éléments d'un corpus, notre méthodologie utilise des techniques de traitement d'images pour détecter les différents types de transformations géométriques ou photométriques expliquant la production des versions altérées d'une image originale. Par la suite, nous avons identifié et visualisé les différentes lignées constituant le corpus. Enfin, nous avons généré un graphe hiérarchique illustrant le réseau de dissémination de chaque image depuis sa version originale jusqu'à ses versions dérivées, en identifiant les transformations entre couple d'images voisines. De plus, pour les images qui ont subi un traitement d'édition, une approche probabiliste est proposée pour détecter et catégoriser les différents types d'édition. Des expérimentations sur un ensemble d'images représentant des copies d'une photographie de *Gustave Doré* ont montré la validité de notre approche.

Notre méthode dans le présent mémoire a marquée un succès pour suivre l'évolution d'un corpus visuel constitué par les différentes copies de la même image circulant sur le web. Des approches futures peuvent être explorées pour améliorer notre approche. Par exemple, on peut dépasser la limite de l'absence des métadonnées par l'analyse de paratexte des différentes pages Web contenant les images. Ce volet s'appuie sur une analyse qualitative du contenu des pages Web sources des images, visant notamment à contextualiser ces images. À partir des URLs, chaque page où l'image originale a été republiée sera identifiée selon une catégorisation typologique (institution muséale, presse, blogue, etc.). Par ailleurs, dans notre présent travail les images racines sont identifiées dès le départ par un expert. L'ajout donc d'une autre phase d'identification des images racines peut être une nouvelle voie de recherche très intéressante pour automatiser la génération du graphe de précedence.

Bibliographie

- [1] R C. Gonzalez et R.E. Woods. *Digital Image Processing*. Prentice Hall, 3rd ed, 2008.
- [2] M. Du, A-C. Sankaranarayanan and R. Chellappa. Robust Face Recognition From Multi-View Videos. *IEEE Trans. Image Processing*, Vol. 23, No. 3, pp. 1105-1117, 2014.
- [3] O. Russakovsky, L-J. Li, F-F. Li. Best of both worlds : Human-machine collaboration for object annotation. *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2121-2131, 2015.
- [4] J. Long, E. Shelhamer, T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3431-3440, 2015.
- [5] S-J. Hwang and K. Grauman. Learning the Relative Importance of Objects from Tagged Images for Retrieval and Cross-Modal Search. *Int'l Journal of Computer Vision*, Vol. 100, No. 2, pp. 134-153, 2012.
- [6] Y-J. Lee, and K. Grauman. Object-Graphs for Context-Aware Visual Category Discovery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 2. pp. 346-358, 2012.
- [7] P. Aigrain, H.Zhang et D. Petkovic. Content-based representation and retrieval of visual media : A state-of-the-art review. *Multimedia tools and applications*, vol. 3, No. 3, pp. 179-202, 1996.

- [8] Adobe, X. M. P. Adding intelligence to media. URL <http://www.adobe.com/products/xmp>.
- [9] A. Alzubi, A. Amira et N. Ramzan. Semantic content-based image retrieval : A comprehensive study. *Journal of Visual Communication and Image Representation*, vol. 32, No. C, pp. 20-54, 2015.
- [10] D. Beer et R. Burrows. Popular culture, digital archives and the new social life of data. *Theory, Culture and Society*, pp. 47-71, 2013.
- [11] G. K. Birajdar, V. H. Mankar. Digital image forgery detection using passive techniques : A survey. *Digital Investigation*, vol. 23, No. 3, pp. 226-245, 2013.
- [12] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler et F. Yergeau. Extensible markup language (XML). *World Wide Web Consortium Recommendation REC-xml-19980210*, vol. 16, 1998.
- [13] S.A. Berrani, L. Amsaleg, and P. Gros, .Robust content-based image searches for copyright protection,. in Proc. of ACM Int. *Workshop on Multimedia Databases*, 2003, pp. 70.77.
- [14] A. Joly, C. Frfelicot, and O. Buisson, .Robust content-based video copy identification in a large reference database,. in Int. Conf. *on Image and Video Retrieval*, 2003, pp. 414.424.
- [15] Y. Ke, R. Sukthankar, and L. Huston, .Efficient near-duplicate detection and sub-image retrieval,. in Proc. of *ACM Int. Conf. on Multimedia*, 2004.
- [16] KIM, Changick. Content-based image copy detection. *Signal Processing : Image Communication*, 2003, vol. 18, no 3, p. 169-184.
- [17] Boyd, C. and Forster, P. 2004. Time and date issues in forensic computing'a case study. *Digital Investigation*, 1, 1,18-23.
- [18] P. Brivio, M. Tarini et P.Cignoni. Browsing large image datasets through Voronoi diagrams. *IEEE Trans. on Visualization and Computer Graphics*, Vol. 16, No. 1, pp. 1261-1270, 2010.

- [19] A. Cornuéjols, L. Miclet, Y.Kodratoff, *Apprentissage Artificiel, Concepts et algorithmes* ISBN 2-212-11020-0 , 2002
- [20] J. E. Camargo, J. C. Caicedo et F. A. Gonzalez. A kernel-based framework for image collection exploration. *Journal of Visual Languages and Computing*, Vol. 24, No. 1, pp. 53-67., 2013
- [21] C. Carson, S. Belongie, H. Greenspan et J. Malik. Blobworld : Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 8, pp. 1026-1038, 2002.
- [22] Casemajor, N. (2016, April). Embedded Metadata and the Digital Lifecycle of Images : *Methodological Challenges*. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pp. 871-872.
- [23] E. Cherhal. Présentation des standards : le Dublin Core (DC). *Communication rédigée par l'auteur suite à sa communication orale* , 2004.
- [24] E. Dagiral et L. Tessier. Explorer les cultures visuelles sur le web. *Les Cahiers du numérique*, pp. 17-35, 2014.
- [25] M. Dewing. Les médias sociaux : Introduction. *Bibliothèque du parlement*, 2010.
- [26] C. Ding et D. Tao. A comprehensive survey on pose-invariant face recognition. *preprint arXiv :1502.04383*, 2015.
- [27] C. Nathalie *Temporalités et spatialités de la migration des contenus sur le Web* z. 2015, Document de travail, non publié.
- [28] G. Gagaudakis and P.L. Rosin. Shape Measures for Image Retrieval. *IEEE Int'l Conf. on Image Processing*, pp. 757-760, 2001.
- [29] R. C. Gonzalez, et W. E. Rechard . "Histogram processing." *Digital Image Processing. 3rd edition*, 2012.

- [30] Y. Gu, C. Wang, J. Ma, R.J. Nemiroff, D.L. Kao et D. Parra. Visualization and Recommendation of Large Image Collections Toward Effective Sensemaking. *Information Visualisation*, doi : 10.1177/1473871616630778, 2016.
- [31] I. Herman, G. Melançon et M. S. Marshall. Graph visualization and navigation in information visualization. *IEEE Trans. on Visualization and Computer Graphics*, Vol. 6, No. 1, pp. 24-43, 2000.
- [32] N. A. Ibraheem, M. M. Hasan, R. Z. Khan et P. K. Mishra. Understanding color models : a review. *ARPJ Journal of Science and Technology*, Vol. 10, pp. 265-275, 2012.
- [33] Y. Ke, R. Sukthankar, L. Huston et R. Sukthankar. Efficient near-duplicate detection and sub-image retrieval. In *ACM Multimedia*, Vol. 4, No. 1, p. 5, 2004.
- [34] R. Kumar, J. Novak et A. Tomkins. Structure and evolution of online social networks. In *Link mining : models, algorithms, and applications*, pp. 337-357, 2010.
- [35] E. LOUBIER, W. BAHOUN et B. DOUSSET. La prise en compte de la dimension temporelle dans la visualisation de données par morphing de graphe. *Colloque Veille Stratégique Scientifique et Technologique*, Marrakech, Vol. 21, 2007.
- [36] L. Manovich, J. Douglass et W. Huber. Understanding scanlation : how to read one million fan-translated manga pages. *Image and Narrative*, Vol. 12, pp. 206-228, 2011.
- [37] L. Manovich et J. Douglass. Visualizing temporal patterns in visual media. *Visualizing the 21st Century*, 2009.
- [38] L. Manovich. How to compare one million images?. *Understanding Digital Humanities*, pp. 249-278, 2012.
- [39] L. Manovich. Media visualization : Visual techniques for exploring large media collections. *Media Studies Futures*, 2012.

- [40] L. Manovich. Style Space : How to Compare Image Sets and Follow Their evolution. *Software Studies Initiative*, 2011.
- [41] C. W. Niblack, R.Barber, W. Equitz, M. D. Flickner, E. H.Glasman, , D. Petkovic et G. Taubin. QBIC project : querying images by content, using color, texture, and shape. *SPIE Symposium on Electronic Imaging : Science and Technology*, pp. 173-187, 1993.
- [42] V. E. Ogle et M. Stonebraker. Chabot : Retrieval from a relational database of images. *Computer*, Vol.28, pp. 40-48, 1995.
- [43] C. Otto, H. Han, et A. Jain. How does aging affect facial components'. *European Conf. on Computer Vision : Workshops and Demonstrations*, pp. 189-198, 2012.
- [44] U. Park, Y.Tong et A. K.Jain. Face recognition with temporal invariance : A 3d aging model. *IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, pp. 1-7, 2008.
- [45] N. Perlin. Introduction to metadata. *IEEE Int'l Conf. on Professional Communication*, pp. 153-155, 2006.
- [46] [http ://free-tools.fr/composants/d3-js-donnees-dirigees-par-documents-3069/](http://free-tools.fr/composants/d3-js-donnees-dirigees-par-documents-3069/)
- [47] M. Petremand, M. Louys, C. Collet et F. Flitti. Représentation couleur d'images multispectrales astronomiques, *Color Display For Multiwavelength astronomical images* , 2004.
- [48] R. Pienta, J. Abello, M. Kahng et D. H. Chau. Scalable graph exploration and visualization : Sensemaking challenges and opportunities. *Int'l Conf. on Big Data and Smart Computing*, pp. 271-278, 2015.
- [49] J. C. Platt, M. Czerwinski et B. AField. PhotoTOC : Automatic clustering for browsing personal photographs. *Information, Communications and Signal Processing*, pp. 6-10, 2003.
- [50] D. Ritendra, D. Joshi, J. Li, J.Z. Wang. Image Retrieval : Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys*, Vol. 40, No. 2, pp. 1-60, 2008.

- [51] K. Schoeffmann, D. Ahlström et C. Beecks. 3d image browsing on mobile devices. *IEEE Int'l Symposium on Multimedia*, pp. 335-336, 2011.
- [52] Sedona Working Group on Electronic Document Retention and Production. 2007. The Sedona Principles (Second Edition) : Addressing Electronic Document Production. The Sedona Conference.
- [53] K. Schoeffmann et D. Ahlström. Similarity-based visualization for image browsing revisited. *IEEE Int'l Symposium on Multimedia*, pp. 422-427, 2011.
- [54] K. Schoeffmann, F. Hopfgartner, O. Marques, L. Boeszoermyeni et J. M. Jose. Video browsing interfaces and applications : a review. *Journal of Photonics for Energy*, 018004-018004, 2010.
- [55] V. K. Singh, M. Gao et R. Jain. Social pixels : genesis and evaluation. *ACM Int'l Conf. on Multimedia*, pp. 481-490, 2010.
- [56] A. W. Smeulders, M. Worring, S. Santini, A. Gupta et R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.22, No. 12, pp. 1349-1380, 2000.
- [57] J. R. Smith et S.F. Chang et Chang. VisualSEEK : a fully automated content-based image query system. *ACM Int'l Conf. on Multimedia*, pp. 87-98, 1997.
- [58] M. J. Swain et D. H. Ballard. Color Indexing. *Int'l Journal of Computer Vision*, Vol. 7, pp. 11-32, 1991.
- [59] B. Vatant. Des métadonnées à la description des ressources Les langages du web sémantique. Métadonnées : mutations et perspectives. *Séminaire INRIA*, pp. 163-194, 2008.
- [60] J. Z. Wang, J. Li et G. Wiederhold. SIMPLicity : Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 9, pp. 947-963, 2001.

- [61] C. Wang, J. P. Reese, H. Zhang, J. Tao et R. J. Nemirow. iMap : A stable layout for navigating large image collections with embedded search. *SPIE Electronic Imaging*, pp. 86540K-86540, 2013.
- [62] C. Wang, J. P. Reese, H. Zhang, J. Tao, Y. Gu, J. Ma et R. J. Nemirow. Similarity-based visualization of large image collections. *Information Visualization*, Vol. 14, No. 3, pp. 183-203, 2015.
- [63] G. Wolberg. Image morphing : a survey. *The visual computer*, Vol. 14, No. 8, pp. 360-372, 1998.
- [64] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky and M. Ward. Semantic image browser : Bridging information visualization with automated intelligent image analysis. *IEEE Symposium on Visual Analytics Science And Technology*, pp. 191-198, 2006.
- [65] S. H. Yen, M. H. Hsieh, C. J. Wang et H. J. Lin. A content-based painting image retrieval system based on adaboost algorithm. *IEEE Int'l Conf. in Systems, Man and Cybernetics*, pp. 2407-2412, 2006.
- [66] H. Zhang, J. Tao, F. Ruan et C. Wang. A study of animated transition in similarity-based tiled image layout. *Tsinghua Science and Technology*, vol. 18, pp. 157-170, 2013.
- [67] G. Zhu, D. Doermann. Logo matching for document image retrieval. *Int'l Conf. on Document Analysis and Recognition*, pp. 606-610, 2009.
- [68] R.J. Nemirow and J.T. Bonnell. Astronomy Picture of the Day. *Bulletin of the American Astronomical Society*, p. 1291, 1995.
- [69] G. Wolberg. Digital image warping. Los Alamitos, *IEEE computer society press*, Vol. 10662, 1990.
- [70] L. D. Stefano, S. Mattoccia, M. Mola,. An efficient algorithm for exhaustive template matching based on normalized cross correlation. *Int'l Conf. on Image Analysis and Processing*, pp. 322-327, 2003.

- [71] J. P. Lewis. Fast Template Matching. *Vision Interface*, pp.120-123, 1995.
- [72] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment : From error visibility to structural similarity," *IEEE Trans. on Image Processing*, vol. 13, no. 4, pp. 600-612, Apr. 2004.