Etude de la confusion des descripteurs locaux de points d'intérêt : application à la mise en correspondance d'images de documents

Emilien Royer

UNIVERSITÉ DE TOULON

École doctorale nº 548 : Mer et Sciences

THÈSE

pour obtenir le grade de docteur délivré par

Université de Toulon

Mention : Informatique

présentée et soutenue publiquement par

Emilien ROYER

le 24 octobre 2017

Étude de la confusion des descripteurs locaux de points d'intérêt : application à la mise en correspondance d'images de documents

Directeur de thèse : **Frédéric BOUCHARA**

Jury

M. Frédéric BOUCHARA,	Maître de conférences HDR à l'Université de Toulon	Examinateur
Mme. Véronique EGLIN,	Professeur à l'INSA de Lyon	Rapporteur
M. Lionel FILLATRE,	Professeur à l'Université de Nice Sophia Antipolis	Rapporteur
M. Thibault LELORE,	Docteur ingénieur de recherche à MyScript	Examinateur
M. Jean-Marc OGIER,	Professeur à l'Université de La Rochelle	Examinateur
M. Marçal Rusiñol,	Chercheur associé à l'Université Autonome de Barcelone	Examinateur

Université de Toulon
Laboratoire LSIS
UMR CNRS 7296, Avenue de l'Université, La valette

Remerciements

D'aucuns apprécieront l'aspect saugrenu de terminer la rédaction d'un ouvrage par l'écriture de sa première page. C'est qu'en plus d'être le point final au travail de Thèse, l'exercice des remerciements est délicat ; on me reproche bien souvent ma trop grande concision et pourtant il m'était difficile de faire tenir ces quelques mots dans ces pages.

Bien entendu, mes premiers remerciements vont à mon directeur de thèse, Frédéric Bouchara, qui a su penser à moi en me proposant de travailler sous sa direction sur ce sujet fascinant. Nombreuses sont ses qualités, allant de sa très grande compétence dans les sujets de recherche qu'il maîtrise, à sa gentillesse et sa volonté de se rendre disponible malgré son emploi du temps lourdement chargé. Si tous les « *chefs* » pouvaient être comme lui, le monde du travail en serait grandement apaisé.
Je tiens aussi à remercier mes deux rapporteurs, Véronique Eglin et Lionel Fillatre, d'avoir accepté de rapporter mes travaux de thèse et pour leurs précieux commentaires qui guideront la suite de mes travaux. Mais aussi mes examinateurs : Thibault Lelore qui m'a permis de démarrer dans de bonnes conditions, Marçal Rusiñol dont les travaux m'ont beaucoup inspiré et enfin je suis touché que Jean-Marc Ogier ait accepté de faire partie de ce jury malgré son emploi du temps particulièrement chargé, j'en profite pour le remercier doublement pour avoir dirigé mes comités de suivi de thèse où ses remarques ont toujours été d'une grande pertinence et ont été garantes de la cohérence de mes travaux lorsque j'avais tendance à m'égarer.

Mes remerciements vont aussi à ma famille et ma belle-famille. Plus particulièrement, mes parents m'ont donné le goût du savoir et ont su m'accorder la liberté et l'autonomie indispensables pour l'épanouissement de ce plaisir d'apprendre. Mon frère sera toujours pour moi un exemple à suivre, de la grande minutie dont il fait preuve, à la capacité qu'il a de s'adapter à n'importe quel sujet en y transposant son souci du travail bien fait. Mes grands-parents pour s'être toujours assurés que je ne manquais de rien, j'ai pleinement conscience de la chance qui est la mienne. Enfin, merci à toi, Blandine, qui partage ma vie depuis maintenant neuf ans pour ta gentillesse sans pareille et ton soutient constant sur lequel j'ai pu m'accrocher pendant les moments les plus difficiles.

La vie du laboratoire aura aussi été tout à fait agréable et propice au bon déroulement de ma thèse. On y croise de nombreux visages souriants avec qui il est plaisant de discuter de sujets divers et variés : Jean-François, Frédéric et Nathalie (particulièrement pour leurs attentions), Nicolas, Eric, Francesca, Dory, Sylvain, Audrey, Elisabeth, Jean-Paul, Hervé, Vincente, Manchun, Joseph (*Sensei !*), Xavier, Ricardo, Sofiane, Christian, Phuong, Jean-Marc (véritable archimage POSIX), Cyril, Julien, Ikhlef, Kheir-eddine... etc. De surcroît, en plus de son exceptionnelle gentillesse, Elisabeth réalise un travail titanesque pour lequel elle n'est que rarement remerciée à la mesure de son mérite, je tiens à le souligner. De la même manière, ayant eu la chance de siéger au conseil de l'école doctorale 548 comme représentant doctorant, il m'importe de remercier son directeur, Yves Blache, pour qui je témoigne du sérieux avec lequel il a dirigé cette jeune école et l'engagement qui était le sien. J'en profite aussi pour remercier doublement Joseph, Christian et Jean-Marc pour ces soirées ludiques qui permettaient de s'échapper temporairement du quotidien.
Dans cette vie universitaire, il y avait bien entendu aussi mes camarades thésards : Régis, Rémi, Cécile, Tania, Céline, les (!) Vincent, Xuan, Victor, Thibault (qui est allé vers

de nouveaux horizons, j'espère que la récolte sera bonne cette année !), Marius, Amine, Gwendolyne et Elodie (les plus jolies du MIO !), Stéphane, Bruno, Camille, Ouazna, Aïda, Tuan, Giang, Diogone et tout ceux que j'aurais pu oublier... Il était agréable de pouvoir échanger et se confier à des collègues dans le même bateau, merci à vous !

Mais avant les compagnons « d'infortune » de thèse, il y avait aussi les collègues de promotion. Je pense à Vincent, le meilleur d'entre nous, Romain, véritable générateur de bonne humeur, Jean pour son amour des sciences qu'il partageait avec moi (il nous aura malheureusement quitté beaucoup trop tôt...), Jonathan, son côté calme, posé et réfléchi apportait un équilibre indispensable à notre groupe, Philippe pour sa motivation et son rire, tous deux hautement communicatifs, Bilal pour sa gentillesse et nos riches échanges, Julien (P), toujours prêt à rire à mes blagues, Julien (V), chef d'orchestre de ce groupe, je n'oublierai jamais nos séances de révisions ensemble ponctuées par le partage de son amour pour la musique.
La bonne ambiance qui reignait au sein de notre petit groupe aura fait de ces cinq années d'études parmi les meilleures de ma vie, je vous en remercie.

Bien entendu, si je garde un si beau souvenir de mes études post-bac, c'est aussi grâce aux très grandes qualités de l'équipe pédagogique du département informatique de l'université de Toulon. Emmanuel, Hervé, Valérie, Marie-Christine, Philippe, Nicolas, Elisabeth, Christian, Joseph, Pascal, Jean-Pierre... C'est avec beaucoup de réticence que je les remercie avec cette simple énumération mais mes tentatives de remerciements individuels se soldaient par un texte d'une longueur absurde par rapport à ce qu'il est coutume de faire. Ils ont su allumer en moi dès la première année cette passion dévorante pour l'informatique qui a métamorphosé le lycéen ordinaire que j'étais auparavant jusqu'à me transmettre aussi le virus de la pédagogie. Je ne leur en serai jamais assez reconnaissant, ils font le plus beau métier du monde.

Pour conclure, considérant la dévotion que je porte à l'instruction, seul véritable acte d'émancipation, il m'est impossible de limiter ces remerciements aux seuls enseignants de mes études post-bac. Tous ceux qui auront eu la charge de faire de moi le citoyen que je suis aujourd'hui, année après année depuis mon entrée sur les bancs de l'école, de la maternelle jusqu'aux bancs de l'université : je leur dédie cette thèse, car elle est l'aboutissement de leur travail.

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1

Introduction générale

« *Mes chers enfants, nous sommes entrés dans un siècle fabuleux où les miracles, ceux nés de la science, seront quotidiens et apporteront de la joie aux plus pauvres, aux plus humbles. Les maisons auront le gaz, la lumière électrique, souvent même le téléphone. Ce téléphone qui fera que d'ici on pourra parler sans se déranger, et sans crier, à des personnes qui habitent Aubagne ou même Aix-en-Provence. Notre vingtième siècle sera un très grand siècle.*

...

et sauvé par l'instruction, chacun aura sa place dans un monde qui respectera tous les hommes. »

Joseph Pagnol - La gloire de mon père, Yves Robert (1990)

1.1 Le document et nous

Avant de commencer, il nous est difficile de débuter l'exposition de notre travail sans prendre le temps de poser les bases de sa réflexion, de ce que l'on pourrait appeler son *pourquoi*. Univerbation de *pro quid*, ce qui précède la chose, l'emploi de ce mot peut sembler maladroit dans le discours scientifique. La science s'intéressant à décrire le *comment* sans s'attarder au *pourquoi*. Mais il ne peut y avoir d'effet sans cause et ce travail serait incomplet sans la justification de son existence. Aussi, attachons-nous plutôt à voir ce qui va suivre comme l'explication des causes qui ont précédé sa réalisation. Loin de nous l'idée d'imiter Russell et Whitehead en rédigeant les principia mathematica de cette discipline ; cette digression, dont le lecteur témoignera qu'elle n'en est pas une, sera fort heureusement très courte.

Ce discours prononcé par Joseph Pagnol dans l'adaptation cinématographique d'Yves Robert (absent dans le roman) du premier tome des souvenirs d'enfance de Marcel Pagnol est empreint du scientisme ambiant de l'époque. Avec l'aisance de ceux qui regardent le passé une fois que les flots de l'histoire ont coulé, nous pourrions presque qualifier ces paroles de doucement naïves alors que la civilisation connaitra ses deux conflits mondiaux les plus meurtriers de son histoire en l'espace de quarante ans.
Et pourtant, comment reprocher à Joseph ce fervent optimisme lorsque même les plus grands scientifiques de l'époque le partagent ? Bien que trente ans plus tard, David Hilbert affirmera avec une certitude inébranlable qu'« il n'y a pas *d'Ignorabimus* en mathématiques » avant de se faire prouver le contraire par Kurt Gödel et son théorème d'incomplétude. Cet espoir peut en réalité s'expliquer facilement par les avancées technologiques majeures amenées en un court laps de temps par la seconde révolution industrielle qui sont, comme Joseph l'appelle, des miracles nés de la science. La question qui se pose est : comment l'Être humain en est-il venu à réaliser de tels « miracles » ?

Ce dernier occupe une position bien singulière dans le monde animal. A bien des égards, il ne brille pas particulièrement par rapport à d'autres espèces. Ses capacités physiques telles que sa vitesse ou sa force brute sont loin de rivaliser avec celles des grands prédateurs. De prime abord, tout laisse porter à croire que dans le système complexe qu'est notre monde il ne soit relegué qu'à un rang secondaire de proie. Et pourtant si aujourd'hui nous sommes capables de faire ce constat et de le lire, c'est bien parce que l'humain est parvenu à se hisser au statut de maître incontesté de son écosystème. Dans ce premier quart du XXIeme siècle, celui-ci semble même parfois trop étroit pour répondre à ses besoins, provoquant ainsi la sixième extinction massive ; celle de l'Holocène. Comment est-ce possible ?
L'humain est un animal profondément social et nous pourrions même arguer qu'il n'est que la réalisation de ses interactions qu'il a avec autrui. Certes, de nombreux animaux vivent en troupeau, en communauté. Mais l'humain est le seul qui excelle à ce point dans cette capacité sublimée par ses autres talents. L'évolution ayant favorisé cette direction il serait même difficilement concevable de s'en passer aujourd'hui ; les rares ermites ayant choisi ce mode de vie sans interactions ont bien accumulé les compétences nécessaires grâce à leur précédente vie en société.

Bien évidemment, ces constructions sociales ne sont pas apparues spontanément à leur niveau d'aujourd'hui, nous pouvons constater un processus itératif dans l'histoire. Pendant près de trois millions d'années, les communautés nomades du paléolithique restent relativement simples et modestes tout comme les échanges entre celles-ci. Le langage parlé est suffisant pour faire circuler l'information dans ces sociétés. Viennent ensuite les premières sédentarisations lors du néolithique où les améliorations de l'industrie, du commerce et de l'agriculture sont propices à des augmentations significatives de populations et donc, de l'accroissement de la complexité des communautés qui s'organisent en villages puis en villes. Celles-ci, plus importantes, permettent à nouveau de décupler le potentiel humain. Mais arrive le moment où la communication orale n'est plus suffisante pour porter le degré de complexité de ces communauté sédentaires ; transmettre le savoir, synthétiser et échanger de l'information. C'est ainsi que les premières formes d'écriture sont apparues en Mésopotamie, dans cette région dite du *croissant-fertile* et particulièrement favorable à l'époque il y a environ 6000 ans. L'écriture est une immense révolution qui va permettre de soutenir le développement de ces premières civilisations. L'information peut être conservée indépendamment des individus, transmise,

etc. Aussi nous pouvons dire que si le langage est le muscle de la société qui a permis de rassembler les humains, le document en est sa colonne vertébrale qui a soutenu son élévation en civilisation. Toutefois, avec plus de 7 milliards d'êtres humains recensés nous sommes bien loin des civilisations sumériennes en termes de complexité, ne serait-ce que par l'inertie qu'engendre de grands ensembles d'individus, et la question se pose de la solidité du socle écriture / document physique là où certains historiens estiment que l'empire romain a disparu en partie à cause de sa trop grande complexité.

Fort heureusement, depuis une soixantaine d'années, un autre « miracle » scientifique a vu le jour : il s'agit de l'informatique. Nous ne pouvons qu'imaginer ce qu'en aurait pensé Joseph tant est grand son potentiel que nous prenons le parti de comparer à celui de l'invention de l'écriture.

1.2 L'informatisation

Contraction entre *Information* et *automatique*, l'informatique est la science du traitement automatisé de l'information. Les anglais lui ont donné le terme de *computer science* qui a le mérite d'y faire apparaitre le mot *science* mais le défaut d'y mettre l'ordinateur. Or, comme le disent Neal Koblitz et Michael Fellows, l'informatique a autant à voir avec les ordinateurs que la cuisine avec les casseroles [Fellows and Koblitz, 2000]. En recherche, ses champs d'études sont nombreux et souvent entremêlés. Pour n'en citer qu'une partie : la cryptographie pour la communication secrète d'information, les bases de données pour son stockage efficace, la classification automatisée d'objets, l'aide à la décision et enfin ce qui nous intéresse plus particulièrement dans notre étude, la vision par ordinateur où de l'information est extraite de l'image.

Les origines de l'informatique remontent jusqu'à plusieurs siècles en arrière avec par exemple les travaux de Blaise Pascal et sa pascaline mais c'est vers la fin des années 60 lors de l'apparition de la micro-informatique que son essor sera significatif et fera progressivement son entrée dans la vie quotidienne. Et nous ne saurions trop insister sur la dimension omniprésente de l'informatique aujourd'hui, que ce soit dans notre environnement immédiat avec montres, téléphones, ordinateurs, voitures, appareils photo et électroménager, etc. mais aussi de façon plus diffuse : fichiers administratifs, de renseignements, études statistiques, votes, sécurité sociale, comptabilité, etc. L'idée d'une république numérique commence même à faire son chemin. Aussi il ne nous parait pas invraisemblable que le monde de demain sera numérique et informatisé ou ne sera pas.

Et pourtant, en dépit des efforts colossaux qui sont fait pour aller dans cette direction, ce n'est pas une tâche aisée de remplacer un monde pluri-millénaires. Même si de nombreux services comme une déclaration d'impots peuvent maintenant être entièrement dématérialisés, le document est toujours une pièce centrale de notre société : d'après l'INSEE, 398 millions de tonnes de papiers sont encore consommées chaque année dans le monde soit environ 80 000 tonnes le temps de lire ces lignes. Cette transition prend parfois des allures kafkaïennes où une simple tâche administrative comme la réalisation d'un passeport passe par les étapes suivantes : téléchargement d'un formulaire numérisé, impression et complétion sur papier de ce dernier et présentation à la mairie où il sera alors numérisé.

C'est ce dernier point qui a toute notre attention. Comment faire transiter un document physique pour l'amener dans le monde numérique ? Il ne s'agit pas uniquement de créer une copie numérique par le biais d'une représentation en image, il faut aller plus loin en récupérant l'information. Deux alternatives sont possibles : la saisie à la main par un opérateur d'une version numérique ou l'acquisition par un appareil de capture, un scanner ou un appareil photographique et le traitement automatisé de l'image pour en extraire l'information associée. C'est bien entendu ce deuxième choix qui nous intéresse ici. Enfin, en raison de la multiplication des *smartphones* dans la vie quotidienne, la capture dite « nomade » de documents avec les conditions que sont les siennes est un problème de moins en moins négligeable. Fort heureusement, la vision par ordinateur n'est pas une branche récente dans la recherche en informatique.

1.3 La vision par ordinateur

Le traitement automatisé de l'information contenue dans une image est présent depuis les débuts de l'informatique moderne, dans les années 70. A l'époque il était question d'imiter le fonctionnement du cortex visuel humain dans son analyse, le problème étant considéré comme surmontable. Malheureusement cette question s'est avérée beaucoup plus complexe qu'il n'y paraissait à l'époque et un peu comme dans le conflit qui oppose Roger Penrose à Paul Jorion sur la question de la reproductibilité de la conscience [Jorion, 1997], il a été découpé en sous-problèmes plus ou moins indépendants les uns des autres tel qu'illustré par la figure 1.1. Nous constatons que le champ d'étude, concernant les points d'intérêt et le calcul des descripteurs, occupe une place central en amont de la discipline. Or, une capture nomade d'un document papier peut avoir les objectifs suivants :

— reconnaissance : l'utilisateur cherche à partir d'une photographie qu'il vient de réaliser, à savoir de quel document provient le papier ciblé en interrogeant une base de données.

— numérisation : l'utilisateur souhaite numériser un document papier. Compte tenu des conditions laborieuses d'acquisition d'une capture nomade, il peut-être nécessaire de réaliser plusieurs captures de la cible. Les images sont ensuites fusionnées en une unique version améliorée par alignement automatique et mosaïquage.

— extraction de forme 3D : étape intermédiaire à la précédente, si le document n'est pas parfaitement plat, il peut être nécessaire d'estimer sa courbure afin de procéder à un redressement.

Ces applications nécessitent toutes la détection de points d'intérêt dans l'image et le calcul des informations qui y sont associées. Il s'agit d'un champ d'étude florissant depuis les débuts de la vision par ordinateur. Les contributions sont nombreuses et rivalisent en ingéniosité et efficacité d'année en année d'autant plus qu'il n'existe pas de solution parfaite au problème, ce qui nécessite parfois de concevoir des algorithmes très spécifiques. Notre chapitre 2 sera consacré à l'étude de cette problématique centrale. Toutefois, comme nous le constaterons dans ce chapitre, une sérieuse complication se pose quant à l'inadaptabilité des contributions proposées jusqu'alors à la question des images de documents.

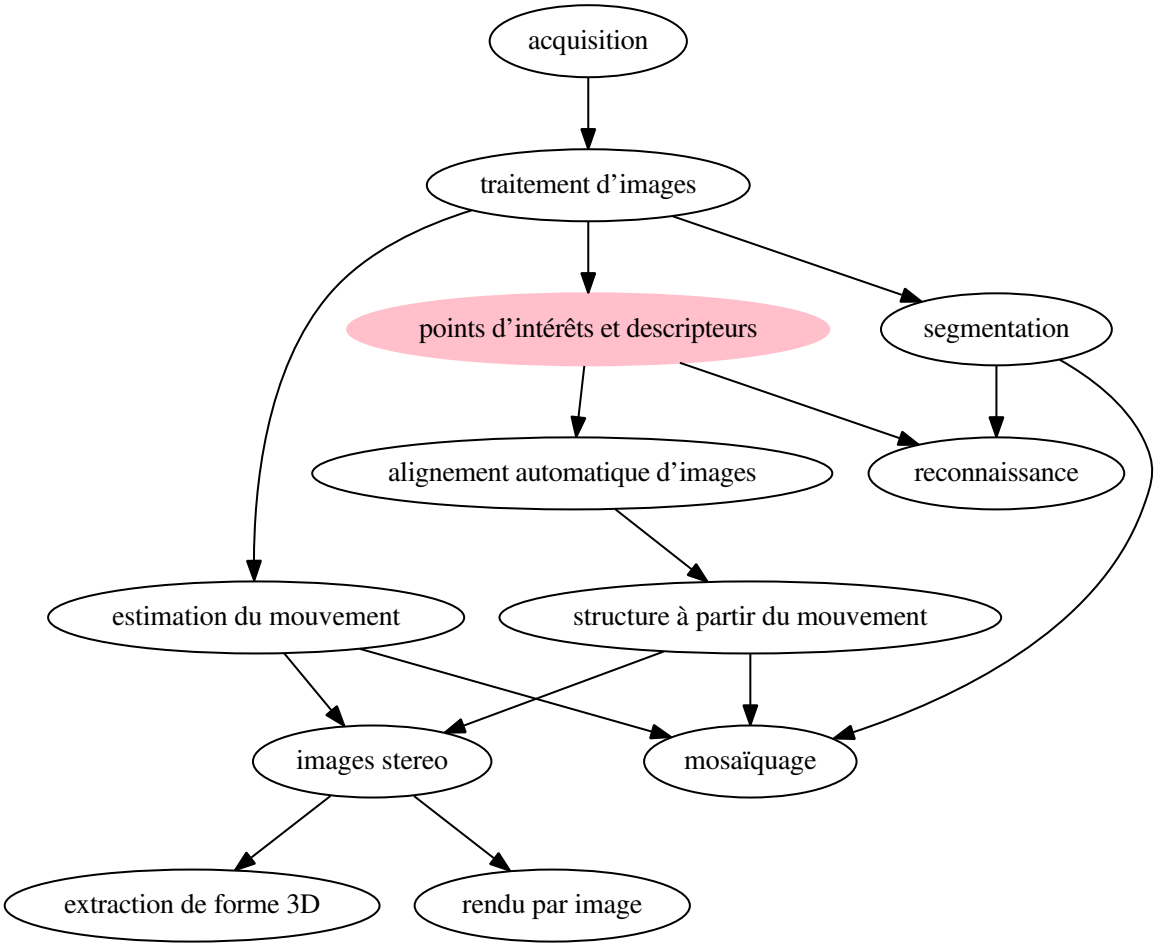


FIGURE 1.1 – Graphe non-exhaustif des dépendances de différents sujets de recherche en vision par ordinateur. La détection de points d'intérêt et le calcul des descripteurs (en rose) occupent une place primordiale.

1.4 Enjeu et problématique

La conception d'algorithmes de détection de points d'intérêt n'a que rarement pris en considération les spécificités des images de documents. Ils ont été principalement conçus pour des images issues du monde réel. Comme nous allons le voir, ils perdent alors des propriétés essentielles et donc en efficacité. La communauté des traiteurs d'images de documents a alors deux possibilités : utiliser ces algorithmes tels quels au risque de subir une perte de performance non négligeable ou redéfinir de nouveaux outils propres aux caractéristiques du document.

Dans cette thèse, nous tentons de concilier les deux mondes en permettant une utilisation des algorithmes traditionnels de détection de points d'intérêt aux images de documents sans ajouter de complications excessives, le tout dans un souci de généricité et de praticité. Comme énoncé précédemment, le chapitre 2 sera consacré à l'analyse des détecteurs de points d'intérêt et du calcul des descripteurs associés afin de mettre en exergue l'inadaptabilité de ceux-ci à la problématique du document. Le chapitre 3 présente notre principale contribution à la réponse de ce problème par le biais d'un système générique de filtrage des points d'intérêt superflus. Le chapitre 4 discute d'une méthode alternative de résolution en guidant l'extraction des points d'intérêt par l'analyse de la structure du document. Enfin, le chapitre 5 sera consacré à la conclusion et aux perpsectives de ces travaux.

Tout au long de nos travaux, nous tentons de nous inscrire dans la problématique de la capture nomade avec les contraintes qui y sont associées comme des images capturées dans des conditions non maitrisées ou des capacités de calculs et de mémorisations réduites par rapport aux ordinateurs de bureau.

Chapitre 2

Détecteurs et points d'intérêt

« *Le vicomte - Attendez ! Je vais lui lancer un de ces traits ! ... (il s'avance vers Cyrano qui l'observe, et se campant devant lui d'un air fat.) Vous... vous avez un nez... heu... un nez... très grand.*
Cyrano, gravement. - Très.
Le vicomte, riant. - Ah !
Cyrano, imperturbable. - C'est tout ?
Le vicomte - Mais...
Cyrano - Ah ! Non ! C'est un peu court, jeune homme ! On pouvait dire... Oh ! Dieu ! Bien des choses en somme...
»

Cyrano de Bergerac (Rostand), Acte I, scène 4

Sommaire

2.1 Détection

Dans ce chapitre, nous abordons l'extraction ou la détection de ce que nous appelons des points d'intérêt. Puis, en deuxième partie nous étudions comment dégager l'information qui y est associée afin qu'elle soit traitée de façon efficace.

De façon informelle, un point d'intérêt est une région de l'image qui se distingue visuellement de façon particulière dans celle-ci, mais ses propriétés diffèrent selon le contexte (le type d'image) et le résultat attendu de l'application qui lui est destinée. Nous pouvons néanmoins présenter les deux plus importantes et répandues, qui sont :

— la discriminabilité : deux points d'intérêt à des endroits différents de l'image doivent le plus possible porter des informations visuelles différentes. Ce critère est particulièrement important et se trouve comme nous allons le voir au cœur de nos travaux,

— la répétabilité : lorsque plusieurs images décrivent la même scène ou le même objet mais avec des propriétés différentes comme le point d'observation, la luminosité, etc., les points d'intérêt détectés dans les différentes images doivent correspondre aux mêmes emplacements dans la réalité physique.

Ces deux critères sont illustrés avec la figure 2.1 et nous allons maintenant discuter des notions de perception visuelle liées à ces points d'intérêt.

2.1.1 Notions de saillance

Décrire de façon rigoureuse le principe et les propriétés visuelles d'un point d'intérêt n'est pas une tâche triviale car l'intégration de cette notion dans notre cerveau créée un biais. Celle-ci est donc par définition intuitive, *i.e.*, c'est une connaissance qui n'est pas le fruit d'un raisonnement. En établissant le constat de la grande performance de notre appareil visuel, s'appuyer sur le processus cognitif de la vision pour construire un détecteur de point d'intérêt semble donc être une approche pertinente. Cependant malgré les dernières décennies de découvertes des chercheurs en neuro-sciences, son fonctionnement reste toujours très mystérieux tant sa complexité est grande.

Jolion *et al.* ont réalisé en 2000 un ouvrage remarquable [Jolion, 2000] sur la question dans lequel ils synthétisent les connaissances de l'époque sur le système visuel humain, de la rétine aux circuits neuromorphiques et discutent de la construction de systèmes de vision dérivés. Il y est expliqué comment le nôtre est composé de couches de neurones hiérarchiquement organisées en aires dénommées V1, V2, V4, le cortex inférotemporal postérieur (PIT) et le cortex inférotemporal antérieur (AIT). Chacune d'entre elles semble posséder une fonction précise. Par exemple, les neurones situés au niveau de la rétine réagissent fortement à la perception de la lumière ; ceux de V1 sont sensibles aux contours présents à certains emplacements du champ visuel (et encore, uniquement selon leurs orientations) et à la fréquence spatiale des motifs présents. Il est intéressant de noter une complexité croissante au fur et à mesure de la progression du traitement de l'information visuelle dans ces aires : ainsi, le cortex inférotemporal est sensible à des stimulus biologiquement importants et probablement sélectionnés par l'évolution sur des millions d'années comme la présence d'un visage humain, etc. Pour réaliser l'ampleur de la complexité de ce système, précisons que le nombre de neurones relais rien que dans l'aire V1 est au bas mot de 500 millions, donc l'établissement d'un connectome (cartographie des connexions neuronales) est largement hors de portée de notre technologie actuelle.

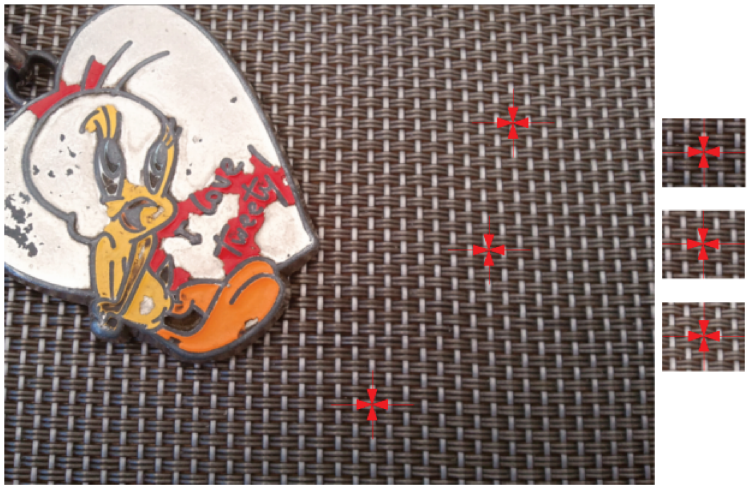


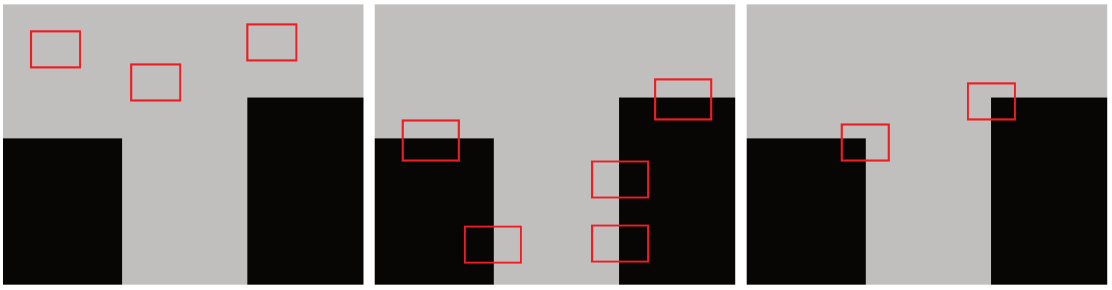
FIGURE 2.1 – Emplacements de trois points d'intérêt dans une image où la présence répétée d'un même motif géométrique réduit grandement leur caractère discriminant ; il est difficile de les différencier si on ne considère que leur voisinage local.

De plus, dans les aires de traitement de haut niveau, les neurones sont de plus en plus difficiles à stimuler car ils répondent à des stimuli de finesse croissante. Et que dire du caractère analogique des transmissions nerveuses à l'opposé du monde discrétisé de nos ordinateurs ? Ainsi, il n'est pas exagéré de dire que le travail titanesque de *retro enginering* de notre système visuel n'en est qu'à ses débuts.

Comme nous sommes forcés de le constater, bien que cette approche soit parfaitement fondée, nous ne disposons pas encore des connaissances requises pour construire un détecteur reproduisant intégralement le processus d'acquisition visuelle humain avec toutes les qualités souhaitées. En revanche, une stratégie béhavioriste, utilisant les propriétés des stimuli bas niveau énoncés précédemment parait être un compromis astucieux, par exemple avec ceux de V1. Si nous reprenons les théories de l'évolution, nous pouvons supposer que si des millions d'années de sélection naturelle ont mis en valeur ces propriétés, c'est bien qu'elles doivent être discriminantes. Commençons par aborder la question en utilisant le problème de l'angle d'ouverture ou de *l'aperture*, comme dans les appareils photos.

Considérons la figure 2.2. Dans chaque sous-figure se trouvent différents points d'intérêt. Dans l'image *(a)*, nous constatons qu'il est impossible de différencier les points, une partie de l'ambiguïté est levée mais persiste dans l'image *(b)* tandis que seule *(c)* permet d'établir une distinction certaine.

Notons que l'information à elle seule de l'emplacement spatial du point d'intérêt dans l'image n'est pas suffisante. Les détecteurs modernes apportent deux compléments importants pour la construction du vecteur caractéristique qui sont l'échelle et l'orientation. Pour en revenir à l'exemple de la figure 2.2, l'échelle caractérise la taille du point d'intérêt ou autrement dit la taille de l'ouverture (la fenêtre rouge sur la figure) et le second donne une mesure de la direction vers laquelle doit être orientée le point d'intérêt pour être analysé. La connaissance de ces deux caractéristiques permet de construire une information qui est invariante au changement d'échelle et à l'orientation.



(a) (b) (c)

FIGURE 2.2 – Le problème de l'ouverture, comme rencontré dans la figure 2.1 ; l'ambiguïté est totalement levée dans le cas (c), lorsque le point d'intérêt est situé sur un coin.

2.1.2 Détections de coins

Mathématiquement, nous pouvons définir un coin comme étant une zone de forte variation locale de l'intensité lumineuse dans deux directions. Une bonne stratégie consiste donc à construire des détecteurs qui recherchent dans l'image les emplacements correspondant aux coins et aux bords en se basant sur cette définition et nous allons passer en revue des contributions significatives.

HARRIS (1988)

Le détecteur de HARRIS [Harris and Stephens, 1988] bien que relativement daté aujourd'hui reste toujours pertinent lorsqu'il s'agit de travailler sur la détection de coins ; il est basé sur des outils mathématiques simples mais efficaces. Comme nous allons le voir plus loin, il est souvent utilisé lors d'une étape intermédiaire dans des détecteurs plus récents.

Partant de la définition d'un coin énoncée juste avant, nous pouvons établir une mesure de la variation d'intensité pour un déplacement (u, v) dans l'image :

$$\mathrm{E}(u,v) = \sum_{x,y} w(x,y) \left[\mathrm{I}(x+u, y+v) - \mathrm{I}(x,y)\right]^2 \tag{2.1}$$

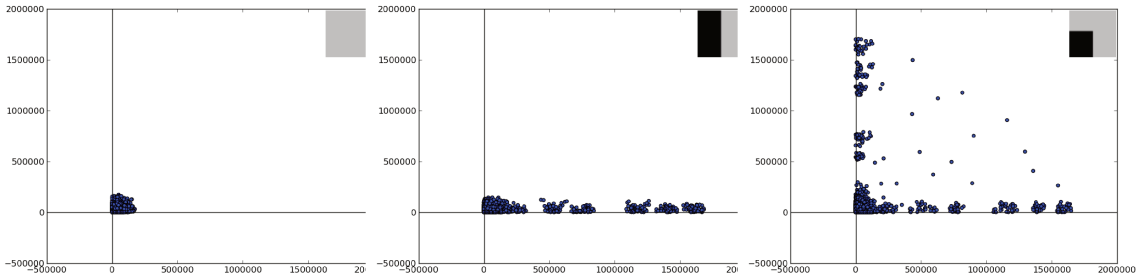
où w correspond à une fonction de pondération, et $\mathrm{I}(i,j)$ à l'intensité lumineuse de l'image à l'emplacement (i,j). Intuitivement, nous constatons que la différence $\mathrm{I}(x+u, y+v) - \mathrm{I}(x,y)$ se rapproche de 0 dans une région relativement stable de l'image et à l'inverse augmente près des coins bords et des coins.

Le but consiste alors à maximiser cette fonction et plus particulièrement le terme de droite. Pour traduire cette expression sous une forme facilement calculable avec un ordinateur, une bonne approche consiste à utiliser les séries de Taylor, pour rappel :

$$f(x+u, y+v) \simeq f(x,y) + u f_x(x,y) + v f_y(x,y) \tag{2.2}$$

où dans notre cas, en notant I_x et I_y les dérivées partielles de I, nous pouvons écrire :

$$\begin{aligned} &\sum \left[\mathrm{I}(x+u, y+v) - \mathrm{I}(x,y)\right]^2 \\ &\simeq \sum \left[\mathrm{I}(x,y) + u\mathrm{I}_x + v\mathrm{I}_y - \mathrm{I}(x,y)\right]^2 \\ &= \sum u^2\mathrm{I}_x^2 + 2uv\mathrm{I}_x\mathrm{I}_y + v^2\mathrm{I}_y^2 \end{aligned} \tag{2.3}$$



(a) Surface plane (b) Bord (c) Coin

FIGURE 2.3 – distribution des dérivées partielles de l'intensité lumineuse en x et y sur une surface plane (a), un bord (b) et un coin (c). Le coin est caractérisé par une dispersion dans les 2 composantes.

Cette expression peut s'écrire sous une forme matricielle :

$$\sum [u,v] \begin{bmatrix} \mathrm{I}_x^2 & \mathrm{I}_x\mathrm{I}_y \\ \mathrm{I}_x\mathrm{I}_\mathrm{Y} & \mathrm{I}_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.4}$$

Soit,

$$[u,v] \left(\sum \begin{bmatrix} \mathrm{I}_x^2 & \mathrm{I}_x\mathrm{I}_y \\ \mathrm{I}_x\mathrm{I}_\mathrm{Y} & \mathrm{I}_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.5}$$

Ramené à l'équation 2.1, nous avons alors :

$$\mathrm{E}(u,v) \simeq [u,v] \mathrm{M} \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.6}$$

où

$$\mathrm{M} = \sum_{x,y} w(x,y) \begin{pmatrix} \mathrm{I}_x^2 & \mathrm{I}_x\mathrm{I}_y \\ \mathrm{I}_x\mathrm{I}_\mathrm{Y} & \mathrm{I}_y^2 \end{pmatrix} \tag{2.7}$$

La figure 2.3 nous montre les distributions des dérivées partielles selon le cas de figure rencontré. Comme il a été postulé antérieurement, un coin correspond à de fortes variations dans les deux directions de l'image. Pour caractériser cette distribution, nous pouvons analyser ses composantes principales en observant les valeurs propres λ_1 et λ_2 de la matrice M. Sachant que le déterminant de la matrice M nous donne le produit $\lambda_1\lambda_2$ et sa trace la somme $\lambda_1 + \lambda_2$, les auteurs de l'article original proposent ainsi le critère final suivant :

$$\mathrm{R} = \det(\mathrm{M}) - \mathrm{K}(\mathrm{trace}(\mathrm{M}))^2 \tag{2.8}$$

avec K un coefficient de pondération empiriquement sélectionné entre 0.04 et 0.06.

Afin de constater la pertinence du critère R, nous pouvons observer à l'aide de la figure 2.4 son évolution en fonction de λ_1 et λ_2 : une valeur élevée signifique que ces deux paramètres sont hauts, une valeur négative correspond à un contour simple et lorsque la valeur absolue est faible il caractérise une surface plane. Pour conclure sur le sujet, c'est la richesse de l'information qu'il porte, associé à sa simplicité qui ont fait le succès de ce détecteur.

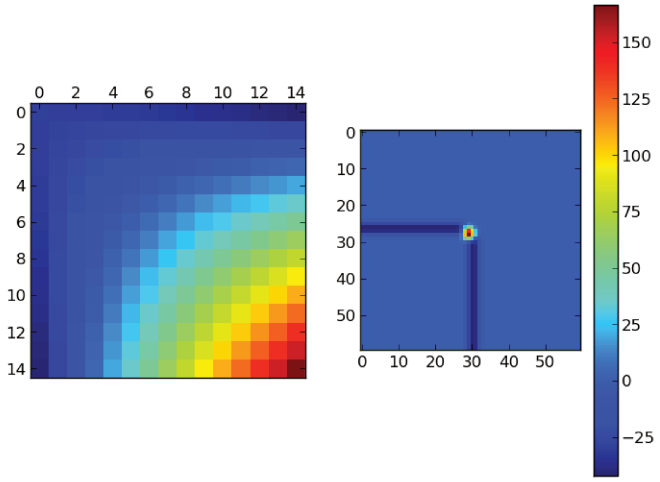


FIGURE 2.4 – Cartographie des valeurs que peut prendre la mesure de Harris en fonction des valeurs de λ_1 et λ_2 (à gauche) avec l'image des mesures de Harris de l'exemple utilisé pour la figure 2.3c (à droite).

SUSAN : Smallest Univalue Segment Assimilating Nucleus (1997)

Au contraire des méthodes inspirées par Harris qui recherchent une dynamique ou évaluent un gradient, SUSAN utilise une approche morphologique qui a inspiré de nombreux autres détecteurs [Smith and Brady, 1997]. Il se décline en détecteur de contours et de coins et c'est cette dernière version qui nous intéresse.

Pour chaque pixel testé, l'algorithme analyse une région circulaire autour de lui en se basant sur son intensité lumineuse. Les pixels à l'intérieur sont alors classés en deux catégories : « similaires » et « identiques ». Intuitivement nous devinons que l'étude du ratio des pixels similaires et différents renseigne sur la morphologie locale de l'image : une surface plane contient presque uniquement des pixels similaires, un contour voit les proportions réparties à peu près équitablement alors qu'un coin présente un taux très haut de pixels différents.

Plus précisément, la région circulaire a pour rayon 3.4 pixels donnant un masque de 37 pixels (le choix de cette dimension résulte d'une approche purement empirique). Il est possible de formaliser la distinction des pixels en classes énoncée ci-dessus en leur attribuant une valeur c :

$$c(\vec{r},\vec{r}_0) = \begin{cases} 1 \text{ si } |\mathrm{I}(\vec{r}-\mathrm{I}(\vec{r}_0)| \leq t \\ 0 \text{ si } |\mathrm{I}(\vec{r}-\mathrm{I}(\vec{r}_0)| > t \end{cases} \tag{2.9}$$

où $\vec{r}_0$ est la position du pixel étudié (le noyau) dans l'image et $\vec{r}$ la position de n'importe quel autre point dans le cercle avec t un paramètre de seuillage. Par la suite, en sommant les réponses c dans le cercle nous obtenons un critère n :

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r},\vec{r}_0) \tag{2.10}$$

Le nombre n correspond à la surface de « l'USAN » (*Univalue Segment Assimilating Nucleus*) et la formalisation de la distinction morphologique s'établit à l'aide du critère sui-

vant :

$$\mathrm{R}(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0) \text{ si } n(\vec{r}_0) < g \\ 0 \text{ sinon} \end{cases} \tag{2.11}$$

avec g un *seuillage géométrique* dont la valeur est fixée par la relation $3n_{max}/4$ dans la détection de contours et exactement $n_{max}/2$ dans la détection de coins, où n_{max} correspond à la valeur maximale que peut prendre n. En pratique toutefois, le calcul de c est lissé en étant remplacé par le calcul suivant :

$$c(\vec{r}, \vec{r}_0) = e^{-\frac{\mathrm{I}(\vec{r})-\mathrm{I}(\vec{r}_0)}{t}} \tag{2.12}$$

Reste de nombreux faux positifs qui peuvent être évités avec l'inclusion de deux tests. Le premier est de calculer le centre de gravité de la surface de l'USAN ; intuitivement nous devinons que dans le cas d'un coin, ce centre de gravité sera détaché du noyau. Le second est particulièrement utile dans les images fortement bruitées, il s'agit de vérifier la continuité de la surface de l'USAN en s'assurant que tous les pixels du masque présents sur une ligne droite entre le noyau et le centre de gravité de l'USAN font bien partie de ce dernier.

Toutefois, appliqué comme tel et même après l'élimination des faux positifs, cet algorithme présente un défaut majeur qui est la multiplication des points retournés dans un voisinage très restreint. Il convient de ne garder qu'un seul point d'intérêt par coin mais celui-ci doit être le même lors d'une modification de l'image, par souci de répétabilité, et ce à une précision au pixel près. Heureusement, ce défaut est facilement contourné en ne conservant dans un voisinage restreint que le candidat dont la réponse USAN est la plus faible.

Algorithme 1 : SUSAN

Données : t : seuil d'intensité
Données : g : seuil géométrique
Résultat : Φ : ensemble de coins détectés
1. Pour chaque pixel, définir une région d'étude circulaire centrée dessus
2. Appliquer l'équation (2.10) pour calculer l'USAN de chacun d'entre eux
3. Appliquer l'équation (2.11) pour construire l'ensemble ϕ des coins candidats
4. Retirer de ϕ les candidats ne passant pas les tests du centre de gravité et de continuité de l'USAN
5. Appliquer la règle du non-maximum pour construire l'ensemble Φ. **retourner** Φ

FAST : Features From Accelerated Segment Test (2006)

Comme l'acronyme le laisse penser, FAST [Rosten and Drummond, 2006] est un algorithme très rapide, c'est l'une de ses principales qualités et le fonctionnement de sa version originale est aussi très simple. Il reprend le principe de SUSAN (d'où l'utilisation du mot *segment*) mais en partant de l'idée qu'il n'est pas nécessaire d'analyser toute la région circulaire autour du pixel étudié mais seulement la bordure.

Autour d'un pixel étudié p ayant pour intensité lumineuse I_p, l'algorithme trace un cercle de Bresenham de 16 pixels ordonnés arbitrairement. Pour chacun d'entre eux, l'intensité lumineuse est comparée aux valeurs $\alpha = \mathrm{I}_p + t$ et $\beta = \mathrm{I}_p - t$ avec t un paramètre de seuillage ; une intensité supérieure à α signifie que ce pixel est plus lumineux que p et inversement dans le cas où elle est inférieure à β, plus sombre. Le test, dénommé AST

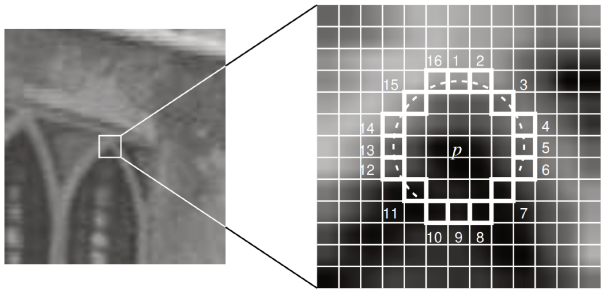


FIGURE 2.5 – Features From Accelerated Segment Test (FAST) ; échantillonnage des pixels autour de celui étudié selon un cercle de Bresenham pour calculer les différences d'intensité lumineuse. [Rosten and Drummond, 2006]

pour *Accelerated Segment Test* est alors de vérifier si un ensemble n de pixels continus est plus sombre / lumineux que p et si tel est le cas, p est considéré comme un coin. Les deux paramètres de contrôle n et t correspondent respectivement au degré d'ouverture maximum du coin à extraire et à la sensibilité du détecteur. Le choix de n est plus délicat qu'il n'en a l'air, il est important d'avoir une valeur permissive pour améliorer la répétabilité mais supérieure à 8 qui correspond au cas des contours. Dans l'article original, n a été empiriquement évalué à 12 (FAST-12) mais c'est la version FAST-9 qui semble être la plus utilisée car présentant un meilleur taux de répétabilité. Enfin, une valeur de t élevée permet de n'extraire que les coins ayant le plus fort contraste de l'image alors que l'abaisser prend en compte des gradients moins élevés.

Ce processus est résumé avec la figure 2.5 et notons qu'il est possible de l'accélérer encore plus en examinant en premier les 4 pixels situés aux points cardinaux du cercle (dans la figure 2.5, 1, 5, 9 et 13) ; si aucun d'entre eux n'est plus sombre (ou plus lumineux) que p, alors ce dernier ne peut pas être situé sur un coin.

Notons que comme avec SUSAN, FAST propose de résoudre le problème de multiplication des candidats spatialements proches en appliquant une règle de suppression des non-maximums ; pour cela les auteurs définissent un critère ν comme étant la somme des valeurs absolues des différences entre p et les pixels du cercle ; lorsqu'il y a multiplication des points retournés par FAST dans un voisinage local, seul celui ayant la valeur ν la plus faible est conservé.

Reste le problème le plus difficile à résoudre : dans quel ordre analyser les différents pixels du cercle ? Assurément, nous pourrions nous contenter de les étudier consécutivement selon un ordre arbitraire afin de rechercher une suite n continue mais l'objectif de la contribution est d'aller vite et les auteurs font remarquer qu'un simple test comme celui des pixels cardinaux présenté précédemment permet de grandement l'accélérer. Trouver l'ordre optimal qui permet de savoir le plus rapidement possible s'il faut continuer le test ou non se ramène à construire un arbre de décision ayant le temps de parcours moyen le moins élevé. Aussi abordent-ils ce problème sous l'angle du *machine learning* en commençant par définir trois classes S de pixels du cercle :

$$S_{p\to x} = \begin{cases} d, I_{p\to x} \leq I_p - t \text{ (plus sombre)} \\ s, I_p - t \leq I_{p\to x} \leq I_p + t \text{ (similaire)} \\ b, I_p + t \leq I_{p\to x} \text{ (plus lumineux)} \end{cases} \tag{2.13}$$

Où p_x correspond à un pixel x considéré selon p. Ils appliquent ensuite la méthodologie de classification supervisée de l'algorithme ID3 pour construire récursivement un arbre

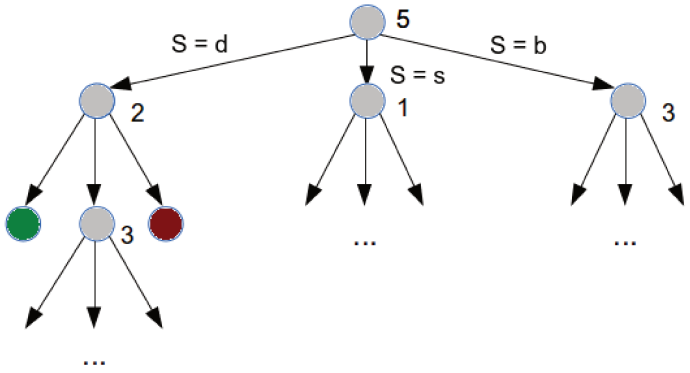


FIGURE 2.6 – Exemple d'arbre de décision ternaire FAST fictif : le numéro à côté de chaque nœud correspond au pixel étudié selon la figure 2.5 et le parcours est effectué selon l'état S. Une feuille verte correspond à un coin et une feuille rouge signifie l'impossibilité d'avoir un test AST positif.

de décision ternaire. A chaque étape d'apprentissage, selon la réponse à la question "*quel est l'état du pixel étudié ?*", l'algorithme sélectionne le prochain pixel à étudier dont la réponse maximise l'entropie de Shannon. Le résultat est un arbre ternaire comme celui présenté par la figure 2.6.

Deux ans plus tard, en 2008, les auteurs proposaient une ultime version de cette amélioration qu'ils nomment trompeusement FAST-ER (car elle ne cherche pas à être plus rapide) pour *Enhanced Repeatability* dont le but est, comme le détail de l'extension de l'acronyme le précise, d'améliorer la répétabilité des points retournés. La principale modification consiste à augmenter le nombre de pixels du cercle en épaississant ce dernier, finalement dans une approche qui rappelle SUSAN si ce n'est que le noyau central de taille $3x3$ n'est pas pris en compte. Évidemment, l'arbre construit est plus grand et l'algorithme final est plus long que FAST original.

AGAST : Adaptative and Generic Accelerated Segment Test (2010)

Les versions de FAST basées sur un arbre sont plus performantes que le test AST d'origine mais elles présentent toutefois le défaut majeur d'être grandement dépendantes des images utilisées lors de la phase d'apprentissage ; un nouvel environnement nécessite de relancer cette dernière et de construire un nouvel arbre. De surcroit, l'ensemble des configurations de pixels n'est pas nécessairement trouvé dans l'arbre et, quand bien même il le serait, le changement de point d'observation de l'image comme lors d'une rotation modifie considérablement la distribution des pixels par ce qui a été appris, ce qui réduit drastiquement la rapidité de l'algorithme utilisant un arbre inadapté.

Les contributeurs qui ont proposé AGAST ([Mair et al., 2010]) souhaitent s'affranchir de ces limitations en construisant un arbre plus performant et qui ne soit pas le fruit d'un apprentissage, d'autant plus que ID3 [Quinlan, 1986] utilisé dans FAST est un algorithme *glouton* qui ne garantit pas un traitement optimal [Hyafil and Rivest, 1976].

La première modification majeure est de réduire l'arité de l'arbre en produisant un arbre binaire, plus facile à optimiser. Au lieu de diverger selon l'état du pixel étudié, ce qui revient à considérer trois questions, les auteurs proposent de ne répondre qu'à une seule dont le choix fait partie de l'étape de construction de l'arbre. Ainsi, alors que la phase d'apprentissage de FAST recherchait à chaque étape quel prochain pixel étudier

selon un des trois états possibles du précédent, AGAST dans la construction de son arbre recherche à la fois le pixel suivant et l'unique question à poser. Répondre à une question binaire permet en plus de considérer d'autres états pour enrichir l'exploration de l'espace de configuration sans impacter l'arité de l'arbre, aussi en ajoutent-ils deux nouveaux :

$$S_{p\to x} = \begin{cases} d, I_{p\to x} \leq I_p - t (\text{ plus sombre}) \\ \overline{d}, I_{p\to x} \geq I_p - t \wedge S'_{p\to x} = u (\text{ Pas plus sombre}) \\ s, I_{p\to x} \geq I_p - t \wedge S'_{p\to x} = \overline{b} (\text{ similaire}) \\ s, I_{p\to x} \leq I_p + t \wedge S'_{p\to x} = \overline{d} (\text{ similaire}) \\ \overline{b}, I_{p\to x} \leq I_p + t \wedge S'_{p\to x} = u (\text{ pas plus lumineux}) \\ b, I_{p\to x} > I_p + t (\text{ plus lumineux}) \end{cases} \tag{2.14}$$

où $S'_{p\to x}$ est l'état précédent, u caractérisant un état inconnu et $\wedge$ une conjonction logique. Pour construire l'arbre, les auteurs utilisent une méthode similaire à [Garey, 1972] en explorant l'espace des configurations possibles partant de la racine de l'arbre de décision où aucun des pixels n'est connu. Les nœuds de l'arbre sont formés en évaluant récursivement le choix d'une question sur un pixel donné jusqu'à ce qu'une feuille soit trouvée (savoir si la configuration actuelle peut toujours ou pas répondre positivement au test AST). Évaluant le coût d'une feuille à 0, celui d'un nœud intermédiaire est obtenu par :

$$\begin{aligned} C_P &= \min_{\{(C_+,C_-\}} C_{C_+} + p_{C_+} C_T + C_{C_-} + p_{C_-} C_T \\ &= C_{C_+} + C_{C_-} + P_p C_T \end{aligned} \tag{2.15}$$

où :

— C_T représente le coût de l'évaluation d'un pixel,

— C_+ et C_- sont les enfants résultants de la réponse à la question du nœud, selon que celle-ci est positive ou négative,

— et P_p, p_{C_+} et p_{C_-} sont les probabilités de la configuration des pixels aux nœuds parents et enfants.

L'introduction du calcul de probabilité est importante pour générer ensuite un arbre « adaptable » et se passer d'une phase d'apprentissage. AGAST différencie deux cas qui produisent des résultats très différents : les surfaces homogènes et les surfaces texturées, fortement hétérogènes. En considérant des probabilités différentes pour les deux, il est possible de construire deux arbres différents. Ensuite, il est possible de les connecter comme montré avec l'exemple de la figure 2.7 lors de l'exécution de l'algorithme ce qui permet de basculer entre les deux arbres selon l'emplacement dans l'image.

2.1.3 Détecteurs avancés

Rechercher les coins dans l'image est une stratégie qui a fait ses preuves mais ce n'est pas la seule. Des approches alternatives consistent à repérer les extremums locaux de réponses à des filtres. Avec un peu de recul, il s'agit toujours d'analyser une dynamique, même si elles paraissent moins intuitives. Nous présentons quelques contributions emblématiques et en profitons pour détailler les stratégies de descripteurs étudiés dans la seconde partie qui sont des améliorations ou des combinaisons de méthodes vues précédemment.

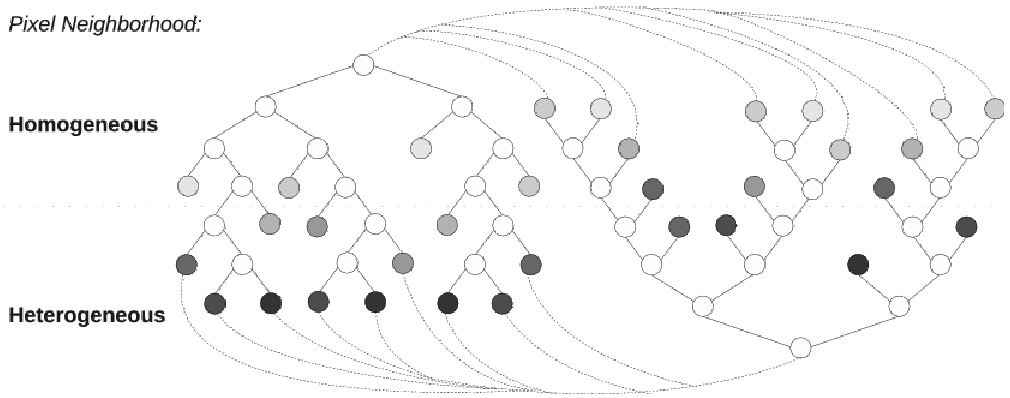


FIGURE 2.7 – Exemple d'arbre de décision binaire AGAST adaptif : deux arbres inter connectés permettent de basculer dans le parcours optimal adapté selon que l'évaluation se déroule en milieu homogène ou hétérogène. [Mair et al., 2010]

SIFT : Scale-invariant feature transform (2004)

Le détecteur SIFT de David Lowe [Lowe, 2004] est une contribution majeure dans le domaine, présentant une invariance à l'échelle et approchant le fonctionnement du cortex visuel humain. Son fonctionnement n'est pas très compliqué : il commence par générer des images progressivement floutées par application de filtres gaussiens en augmentant la variance σ^2 à chaque étape. Le pixel à la position x, y dans l'image I a un équivalent flouté défini par :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.16}$$

Avec G le filtre gaussien suivant :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2.17}$$

Ce processus est répété sur plusieurs « octaves », des variantes en taille de l'image d'origine. Cela permet de prendre en compte plusieurs niveaux de détails. En principe, le nombre d'octaves et d'échelles peut dépendre de la taille de l'image mais Lowe recommande respectivement 4 et 5. Une fois ces images générées il est alors procédé aux calcul des différences des échelles successives dans chaque octave (différences de gaussiennes). Ces calculs sont en réalité une approximation du Laplacien de convolutions Gaussiennes (LoG), pratique pour détecter des zones de fortes variations dans l'image. Pour chacune de ces nouvelles images, on parcourt les pixels un à un en inspectant les 8 voisins dans l'échelle courante ainsi que dans les échelles adjacentes (pour un total de 26 pixels voisins) : si le pixel étudié a l'intensité la plus petite ou la plus grande de cet ensemble de voisinage, il est marqué comme un point d'intérêt. Lowe propose ensuite de faire une recherche sub-pixelique du maximum local (puisque le pixel est en réalité une approximation de ce maximum) par le biais d'une expansion de Taylor de l'image autour du pixel.

Enfin, les points d'intérêt candidats sont sélectionnés selon un critère de saillance qui est l'intensité du contraste local : si celle-ci est en dessous d'un certain seuil, le point d'intérêt n'est pas conservé. Il en va de même pour les pixels situés sur un bord. Une mesure similaire à celle de Harris par l'étude des gradients verticaux et horizontaux permet de

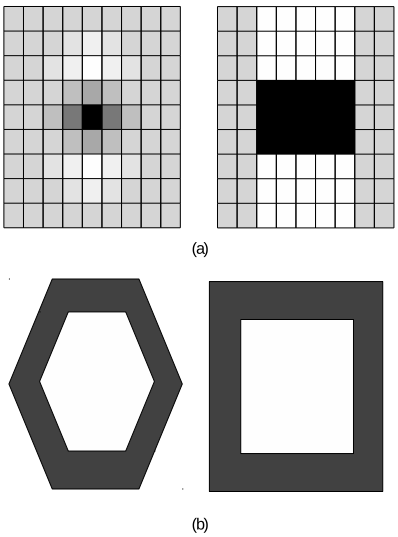


FIGURE 2.8 – (a) à gauche, discretisation d'un noyau gaussien et son approximation par filtre rectangulaire à droite. (b) exemples de filtres à deux niveaux utilisés par CenSurE pour approximer le laplacien.

sélectionner ceux qui sont proches d'un coin. Notons en défaut qu'il n'y a aucune garantie sur une fréquence spatiale équitablement répartie des extremums dans l'espace des échelles, c'est même souvent le contraire qui se produit si des zones regroupent de nombreux points dans un espace géographique réduit.

SURF : Speeded Up Robust Features (2006)

SIFT a été un véritable succès dans la communauté vision mais son coût calculatoire élevé était un défaut contraignant. La contribution de SURF [Bay et al., 2008] visait à répondre à cette problématique en présentant un détecteur et un descripteur quasiment aussi performant mais beaucoup plus rapide à calculer. Dans les grandes lignes, cet algorithme est très similaire à SIFT : il s'agit d'approximer le Laplacien de convolutions de Gaussiennes pour détecter les extremums dans l'espace-échelle. Mais les auteurs font remarquer qu'il est possible d'aller encore plus loin dans l'approximation en remplaçant les convolutions de filtres gaussiens par des filtres rectangulaires, comme illustré par la figure 2.8. L'intérêt d'utiliser des filtres rectangulaires est qu'ils peuvent être calculés très rapidement grâce au principe des images intégrales [Crow, 1984]. Dans les faits, une telle approximation se révèle suffisante.

CenSurE : Center Surround Extremas for Realtime Feature Detection (2008)

Ce dernier algorithme ([Agrawal et al., 2008]) et ses dérivés comme STAR où SUSurE [Ebrahimi and Mayol-Cuevas, 2009] reprend les mêmes idées que SIFT et SURF dans l'exploration espace-échelle en approximant le Laplacien de convolutions de Gaussiennes. L'approche est similaire à celle de SURF avec convolutions de filtres simples à deux niveaux (figure 2.8) mais qui diffèrent de cet algorithme. Les auteurs proposent d'appliquer 7 échelles différentes de ces filtres sur chaque pixel. Comme SIFT, il faut ensuite rechercher les *maxima* locaux dans l'espace-échelle dans un voisinage $3x3x3$ en rajoutant un seuillage empirique pour éliminer les réponses faibles qui risquent de ne pas être retrouvées à nouveau. Un avantage sur SIFT est que les réponses sont calculées sur l'image ori-

ginale, il n'est donc pas nécessaire de réaliser une opération de recherche sous-pixelique par interpolation. Les réponses sont ensuite classées comme pertinentes ou non selon une simple mesure de HARRIS, les coins étant prioritaires.

Oriented FAST (2011)

Oriented Fast est le détecteur utilisé par le descripteur ORB [Rublee et al., 2011] (que nous étudierons dans la section suivante) ; il est apprécié pour sa simplicité et son efficacité, un duo toujours gage de popularité. La stratégie de détection employée part du constat qu'utiliser FAST pour la rapidité de son exécution est une bonne idée mais présente deux défauts que sont le nombre importants de réponses sur des contours et le manque d'information sur l'orientation.

D'après les auteurs le premier point est une source importante de perte du pouvoir discriminant, ce qui est compréhensible si nous nous ramenons au problème de l'ouverture comme présenté précédemment avec la figure 2.2. Pour contourner ce problème ils proposent une solution qui s'avère simple mais efficace en ordonnant les réponses de FAST avec une mesure de Harris. Ainsi, dans l'image étudiée, le seuil t du premier algorithme est suffisamment abaissé pour obtenir plus de N points d'intérêt candidats qui sont alors triés avec la mesure R de Harris afin de sélectionner les N premiers.

L'information de l'orientation est obtenue avec une mesure proposée par Rosin dans [Rosin, 1999] qui est l'intensité du centroïd. À partir des mesures d'intensité lumineuse, on définit un centre de gravité du coin. Le centroïd étant rarement confondu avec ce dernier, le vecteur du décalage peut-être utilisé pour dégager l'information d'une orientation. Rosin établit le moment d'ordre (p,q) d'une surface comme étant :

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

avec $p,q \in \{0,1\}$. Le centre de gravité est alors à l'emplacement C :

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

Soit O l'emplacement du coin, nous pouvons construire le vecteur $\vec{\mathrm{OC}}$. Son orientation est obtenue par :

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

Les auteurs de ORB proposent ensuite d'améliorer l'invariance de cette mesure en restreignant x et y dans une région circulaire de rayon r correspondant à la taille de la région étudiée.

Le principal défaut de ce détecteur est de laisser à l'utilisateur le choix du paramètre N, le nombre de points à extraire. La pertinence de celui-ci peut varier fortement selon le type d'image étudié et nécessite donc la mise en œuvre d'heuristiques associées.

BRISK / FREAK (2011)

Les descripteurs BRISK et FREAK ([Leutenegger et al., 2011, Alahi et al., 2012]) que nous étudierons dans la prochaine partie empruntent l'analyse dans l'espace des échelles que nous venons de voir pour l'appliquer au très performant AGAST et rendre la détection invariante à l'échelle en recherchant la valeur FAST maximale comme mesure de saillance.

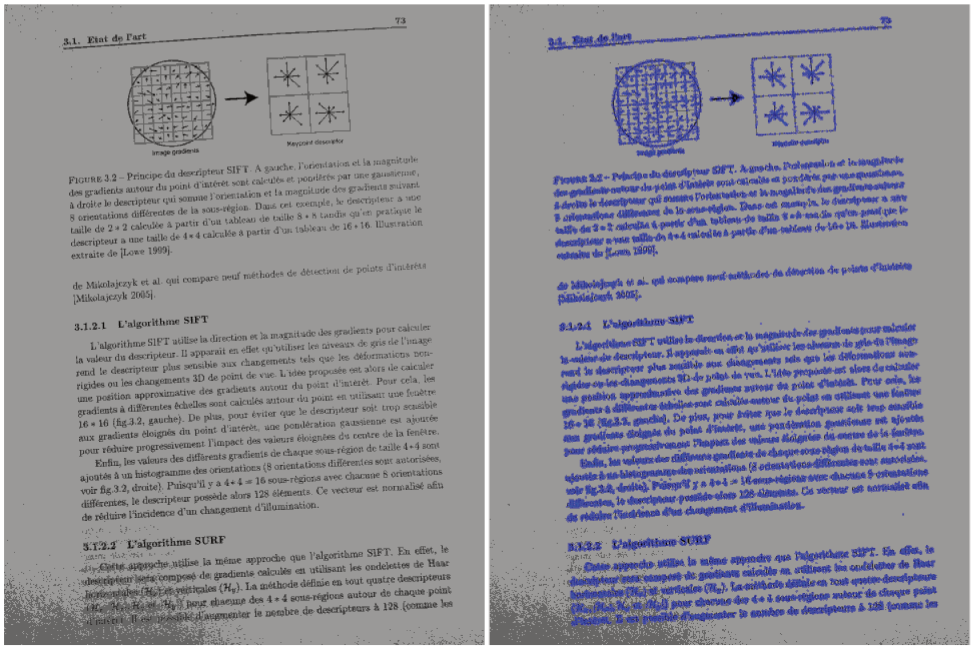


FIGURE 2.9 – Application du détecteur SIFT avec ses paramètres par défaut sur une image de document capturée avec un appareil photographique. A droite, les points d'intérêt retournés par SIFT, environ 20000, sont en bleu.

2.1.4 Conclusions

La détection de points d'intérêt est un sujet de recherche majeur et toujours actif dans la communauté Vision par ordinateurs. Les scènes issues du monde réel sont souvent « pauvres » en dynamique, dans le sens où, rapporté au nombre total de pixels dans une image, ceux manifestant une forte variation dans l'image des gradients ou des traits fortement anguleux (qui sont souvent artificiels), sont minoritaires. Il n'est donc pas étonnant de voir que les méthodes que l'on trouve dans la littérature (voir le tableau 2.1 qui résume les descripteurs étudiés précédemment) cherchent à détecter ces caractéristiques.

Mais qu'en est-il des images de documents ? Comme le montre la figure 2.9, des fortes variations dans l'image des gradients, des zones fortement contrastées, des angles droits et des coins ne correspondent pas à un nombre de pixels minoritaires dans une image de document, c'est même le contraire. Nous constatons donc que ces approches sont inadaptées en raison de leur stratégie de recherche de zones de fort contrastes ou de dynamiques supposées rares dans les images naturelles. Mais avant d'affirmer que le problème est réel, intéressons nous à la façon dont la communauté résume l'information liée aux points d'intérêt par le biais des descripteurs locaux.

Nom	Année	Type	Avantages	Inconvénients
Harris	1988	Détection de coin par analyse de gradient	Simple et efficace, souvent utilisé aujourd'hui comme étape intermediaire	Dépassé, rarement utilisé seul aujourd'hui
SUSAN	1997	Détection de coin par analyse morphologique	Alternative à Harris	dépassé
SIFT	2004	Recherche de maximums dans l'espace-échelle	Contribution majeure, très efficace	Coût algorithmique élevé
FAST	2006	Détection de coin par analyse morphologique	Amélioration de SUSAN, rapide	Apprentissage, résultats non optimal
SURF	2006	Recherche de maximums dans l'espace-échelle	Plus rapide que SIFT	Beaucoup trop de réponses dans les images de documents
CenSurE	2008	Recherche de maximums dans l'espace-échelle	Analyse plus poussée que SURF	Mesure de Harris pour trier les points, peu adapté pour du document
AGAST	2010	Détection de coin par analyse morphologique	Amélioration de FAST	Complexe à implanter
Oriented Fast	2011	Détection de coin par analyse morphologique	Fast avec un calcul d'orientation, bon compromis	Nombre de points à retourner fixé par un seuillage de Harris
BRISK	2011	Mixte	Très bons résultats	Relativement long pour une approche SUSAN-inspirée en raison de l'exploration de l'espace-échelle

TABLEAU 2.1 – Tableau récapitulatif des différents détecteurs de points d'intérêt étudiés, triés par ordre chronologique.

2.2 Descripteurs flottants

Nous appelons descripteurs flottants ceux dont le vecteur caractéristique qui porte l'information n'est pas constitué d'une chaîne de bits. Même si, comme la figure 2.13 le montrera, il n'est pas possible concrètement de faire visuellement la différence car celle-ci est de taille : dans les descripteurs flottants un bit n'a que le sens qu'on lui prête pour coder un nombre tandis que pour les descripteurs binaires celui-ci est une information à part entière. Une autre façon de le dire, peut-être plus simple, c'est que chaque dimension du vecteur caractéristique est un nombre rationnel. Notre travail s'est principalement intéressé aux descripteurs binaires pour des raisons de temps de calculs, aussi nous serons relativement brefs sur la discussion des descripteurs flottants.

Nous choisissons de séparer la présentation des descripteurs flottants en deux classes, ceux se basant sur la construction d'histogrammes et ceux n'en utilisant pas.

2.2.1 Méthodes à histogrammes

Les méthodes à histogrammes réalisent des mesures d'informations reconnues comme étant discriminantes dans une image et répartissent les résultats dans un histogramme selon un critère pertinent ; il peut s'agir d'orientations de gradients ou d'emplacements géographiques de certains pixels dans l'image. Lorsque l'algorithme a terminé de construire l'histogramme, ce dernier est utilisé comme vecteur caractéristique. Nous présentons quelques contributions parmi les plus emblématiques.

SIFT : Scale Invariant Feature Transform (1999)

Proposé en 1999 par Lowe *et al.* [Lowe, 2004], le descripteur SIFT est, tout comme le détecteur, une contribution emblématique de cette discipline parmi les plus influentes.

Autour du point d'intérêt choisi, l'algorithme divise la région d'étude en $n \times n$ sous régions de taille 8×8 ($n = 4$ le plus souvent). Pour chaque sous région, il construit un histogramme des orientations du gradient calculé sur chaque cellule. Cet histogramme est divisé en 8 entrées correspondant à des itérations d'angle $\frac{\pi}{4}$ d'orientation et chaque ajout est pondéré en fonction de la magnitude du gradient. Ainsi, dans le cas traditionnel où $n = 4$, la surface d'étude est divisée en $4 \times 4 = 16$ régions, soit 16 histogrammes de 8 entrées : le vecteur descripteur est donc composé de 128 nombres entiers. Le processus est illustré par la figure 2.10.

Dans l'objectif de privilégier les informations proches du point d'intérêt et d'atténuer les déplacements dans l'espace des descripteurs (dans le cas où la fenêtre d'étude se déplacerait légèrement à cause d'une imprécision lors la construction du point d'intérêt après variations de l'image) il est proposé de recourir à une variable d'ajustement sous la forme d'une fonction gaussienne centrée sur le point d'intérêt, dont l'écart-type σ est égal à la moitié de la taille de la fenêtre d'étude.

L'invariance au changement d'échelle et à l'orientation se fait à l'aide des informations retournées lors de l'étape de détection. Ainsi, les orientations du gradient subissent une rotation qui est fonction de l'orientation estimée lors de la phase de détection et la taille

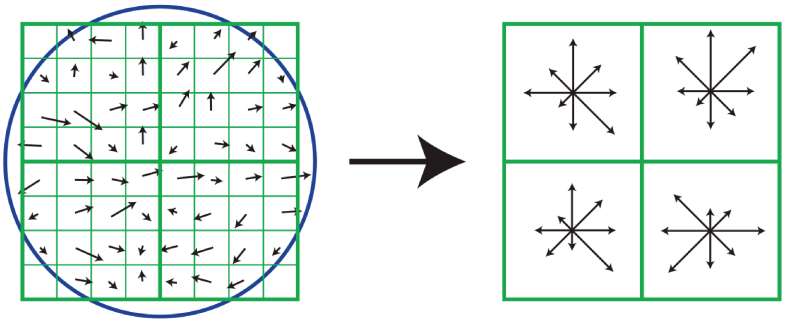


FIGURE 2.10 – Synthèse de la construction du vecteur descripteur SIFT dans le cas où $n = 2$. L'étude du gradient local permet de construire un histogramme des gradients pour les $n \times n$ sous régions. [Lowe, 2004]

de la fenêtre d'étude est le produit d'une constante par la taille de l'échelle du point d'intérêt. L'algorithme propose aussi une résistance au changement d'illumination en normalisant le vecteur. Ainsi, une modification du contraste qui multiplierait chaque intensité de gris de l'image par une constante multiplierait tous les gradients par cette même constante, la conséquence est ainsi annulée par la normalisation.
Un cas problématique restant est une modification non linéaire et partielle de l'éclairage de l'image qui ne garantit plus la répétabilité du descripteur.

Shape Context (2000)

Dans cette approche [Belongie et al., 2002], les auteurs conçoivent les formes des objets comme étant des sous-ensembles fini situés sur les contours d'ensembles de points potentiellement infinis. Ces sous-ensembles peuvent être obtenus à l'aide d'algorithmes classiques comme celui de Canny, et n points y sont échantillonnés à intervalles réguliers (bien que les auteurs précisent que la contrainte de régularité n'est pas obligatoire).

Le descripteur résultant de cette idée est l'étude de la distribution de ces points autour du point d'intérêt concerné. Pour cela, il est proposé d'utiliser un histogramme log-polaire de l'emplacement des points échantillonnés, comme illustré avec la figure 2.11. La définition formelle de cet histogramme à k cellules pour un point d'intérêt p_i est donc :

$$._i(k) = \#\{q \neq p_i : (q - p_i) \in \text{cellule}(k)\} \tag{2.21}$$

L'invariance au changement d'échelle se fait en basant la taille géométrique de l'histogramme comme étant fonction de l'échelle du point d'intérêt et en normalisant les distances $(q - p_i)$ par la distance moyenne α entre toutes les paires de points locales. De plus, la résistance à l'orientation peut-être obtenue en prenant en compte l'angle de la tangente du contour.

Le vecteur descripteur étant un histogramme, les auteurs proposent d'approcher le problème de mise en correspondance avec une mesure du χ^2. Soit $g(k)$ l'histogramme normalisé de l'emplacement i, et $h(k)$ à l'emplacement j :

$$C_S = \frac{1}{2}\sum_{k=1}^{K}\frac{[g(k) - h(k)]^2}{g(k) + h(k)} \tag{2.22}$$

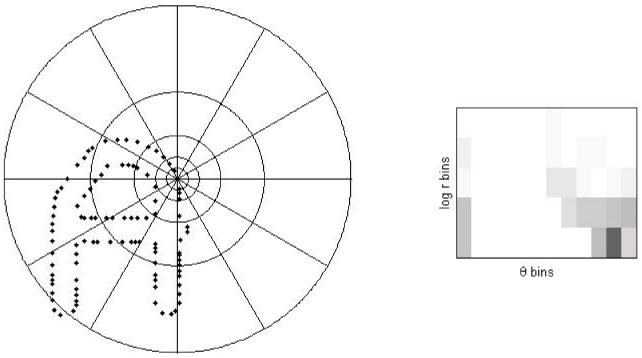


FIGURE 2.11 – Construction d'un histogramme log-polaire *shape context.* [Belongie et al., 2002]

S'ajoute à cela, une mesure liée à l'orientation :

$$C_A = \frac{1}{2}\left|\begin{pmatrix}\cos(\theta_1)\\ \sin(\theta_1)\end{pmatrix} - \begin{pmatrix}\cos(\theta_2)\\ \sin(\theta_2)\end{pmatrix}\right| \tag{2.23}$$

La mesure complète de distance est alors :

$$C = (1-\beta)C_S + \beta C_A \tag{2.24}$$

Avec β un paramètre de pondération mais les auteurs ne donnent pas d'informations quant au choix de ce paramètre.

Présentant une résistance remarquable aux déformations géométrique que peuvent subir les objets aux traditionnels problèmes de bruit, cette méthode s'est ainsi avérée être particulièrement efficace pour la reconnaissance de formes comme les logos ou les lettres et elle est donc fréquemment employée dans les logiciels d'OCR mais peut aussi être appliquée à d'autres problèmes. Ainsi, a été proposée, une extension aux problèmes de reconnaissance d'objets 3D où l'histogramme log-polaire devient une sphère [Frome et al., 2004].

PCA-SIFT (2004)

Comme son nom le laisse supposer, PCA-SIFT [Ke and Sukthankar, 2004] est une application de l'analyse en composantes principales (ACP) [Pearson, 1901] à l'algorithme SIFT et plus précisément sur l'image des gradients calculée autour du point d'intérêt. Pour résumer très sommairement, l'ACP est une transformation linéaire qui réalise une projection dans un nouvel espace où la variance maximale des données observées sert de nouvel axe de projection, que l'on nomme la première composante principale. Il en va ainsi pour les axes suivants en gardant à l'esprit que ceux-ci doivent être orthogonaux. C'est une méthode très efficace pour réduire le nombre de dimensions et c'est l'objectif de cette modification de SIFT.

La première étape de cet algorithme est le calcul d'un espace propre servant à projeter les vecteurs dans un nouvel espace de dimensions réduites. Pour cela, les auteurs proposent de calculer à part les gradients horizontaux et verticaux dans une région de taille 41×41 autour du point d'intérêt (la taille réelle en pixels est bien entendu fonction de l'échelle du point d'intérêt détecté), ce qui donne un premier vecteur descripteur de 3042 éléments et comme d'habitude, celui-ci est normalisé pour résister aux changements d'illumination. Cette opération est répétée sur un ensemble d'images de test totalisant 21000 points d'intérêt. Au final, une ACP est appliquée à la matrice de covariance des vecteurs obtenus et la matrice ayant les plus grands n vecteurs propres est sélectionnée.

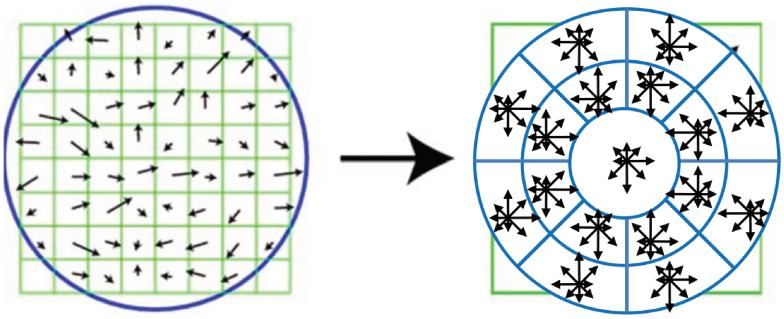


FIGURE 2.12 – Décomposition de la région d'étude par GLOH (à droite), on constate une représentation spatiale différente de SIFT (à gauche) ainsi qu'une répartition log-polaire. [Mikolajczyk and Schmid, 2005]

Passée cette phase d'apprentissage, l'algorithme est très simple. Les vecteurs descripteurs sont construits de la même façon qu'expliqué ci-dessus et sont projetés dans un espace à l'aide de l'espace propre obtenu précédemment. Le paramètre n a été mesuré empiriquement par les auteurs et correspond à $n = 20$. Ainsi, le vecteur descripteur final est de dimension 20 seulement au lieu de 128 pour SIFT.

GLOH : Gradient Location-Orientation Histogram (2005)

Le mode opératoire de GLOH [Mikolajczyk and Schmid, 2005] est similaire en tous points à SIFT si ce n'est le découpage rectangulaire de la région d'étude qui est remplacée par un partitionnage log-polaire (figure 2.12). L'histogramme résultant possède 272 entrées, ce qui est supérieur à celui utilisé par SIFT (128). Aussi, les auteurs ont recours à la même méthode utilisée un an plus tôt par PCA-SIFT en réduisant la dimension à 128 à l'aide d'une analyse en composantes principales.

2.2.2 Méthodes sans histogrammes

L'utilisation des histogrammes est relativement intuitive pour caractériser une information et comme nous avons vu ils présentent l'avantage de permettre l'utilisation des outils de la théorie de la statistique lors des calculs de distance par exemple. Cependant, il n'est pas nécessaire d'avoir recours à des histogrammes pour calculer un descripteur. Nous présentons la contribution la plus emblématique qui réalise le calcul d'un descripteur sans avoir à dénombrer des quantités dans un histogramme.

SURF : Speeded-Up Robust Features (2004)

Cet algorithme [Bay et al., 2008] est très souvent comparé à SIFT en raison de la similarité de son approche et de ses résultats avancés comme étant comparables, de surcroît, il est plus rapide dans son exécution.

Le processus débute de la même façon qu'avec SIFT : un voisinage d'étude autour du point d'intérêt choisi est découpé en sous-régions et ici aussi, il s'agit le plus souvent d'une grille 4×4. Sa taille est fonction de l'échelle s retournée par le détecteur ($20 \times s$ dans l'article original). En revanche, SURF ne construit pas d'histogramme de gradient avec les sous-régions mais calcule des sommes d'ondelettes de Haar, verticales $\mathscr{H}_y$ et horizontales $\mathscr{H}_x$. Localement, le vecteur construit, composé de quatre éléments, est :

$$\nu = \{\sum \mathcal{H}_x, \sum \mathcal{H}_y, \sum |\mathcal{H}_x|, \sum |\mathcal{H}_y|\} \tag{2.25}$$

Ramené sur l'ensemble du voisinage d'étude, le vecteur descripteur a une dimension de $4 \times 4 \times 4 = 64$ éléments. Toutefois, habituellement c'est une variante étendue qui est utilisée pour ajouter du pouvoir discriminant : les sommes de $\mathcal{H}_x$ et $|\mathcal{H}_x|$ (resp. $\mathcal{H}_y$ et $|\mathcal{H}_y|$) sont calculées séparément en fonction du signe de $\mathcal{H}_y$ (resp. $\mathcal{H}_x$). Ceci double le nombre d'éléments et permet de construire des vecteurs de dimension 128 tout en préservant la complexité de l'algorithme puisque celui-ci se base sur l'utilisation des images intégrales.

Une dernière amélioration utilisée lors de la phase d'appariement est la prise en considération du signe du laplacien, la trace de la matrice hessienne, sur le point d'intérêt : celui-ci indique si la zone étudiée correspond à une région claire sur une surface sombre ou l'inverse. Cette approche permet d'éviter de calculer les distances entre deux descripteurs de points qui ne peuvent pas correspondre, réduisant ainsi le temps d'appariement, et d'empêcher un faux positif qui correspondrait à une inversion du contraste mais présentant les mêmes dynamiques de gradients. Notons que le recours à cet opérateur n'est source d'aucune pénalité par rapport au temps d'exécution puisque le laplacien est déjà calculé au préalable lors de la phase de détection.

2.3 Descripteurs binaires

Pendant longtemps, la principale préoccupation - légitime - des descripteurs a été d'améliorer le pouvoir discriminant de l'information extraite du point d'intérêt en caractérisant précisément sa particularité. Le faible coût calculatoire, bien qu'étant une caractéristique appréciable de l'algorithme, était secondaire étant donné le besoin peu important de processus de traitement en temps réel. L'arrivée et la démocratisation de l'informatique embarquée a toutefois mis en avant ce besoin, l'utilisateur ayant besoin d'un retour rapide, si possible en temps réel.

Une classe de descripteurs particulièrement bien adaptée à cette problématique est la famille des descripteurs binaires. Cette appellation vient du fait que les éléments du vecteur caractéristique ne correspondent pas à la mesure d'une quantité (comme peut l'être un histogramme de gradients) mais à la réponse à une question : celle-ci étant soit négative, soit positive, on lui attribut respectivement 0 ou 1. Le vecteur est donc une suite de bits, ce qui est évidemment aussi le cas d'un descripteur flottant : comme le montre la figure 2.13, il n'y a dans la forme pas de différence. Toutefois, là où dans l'exemple montré le premier élément du vecteur SIFT correspond à la quantité 2, son équivalent 181 pour BRIEF est en réalité la concaténation de 8 éléments, à savoir : 10110101 ; Les bits portent indépendamment et individuellement une information.

Le gain est très souvent triple : le calcul du vecteur est en règle général plus rapide, reposant sur des tests simples (très souvent, une différence d'intensité de gris entre deux pixels). Les vecteurs descripteurs sont généralement d'une taille plus faible (256 bits pour BRIEF contre 1024 pour SIFT). Enfin, un avantage particulièrement intéressant réside dans la phase de mise en correspondance : soient deux vecteurs descripteurs X et Y, possédants n composantes x/y_i, $i \in 1...n$. Ceux appartenant à la catégorie des descripteurs flottants utilisent pour cela une mesure de distance euclidienne (équation 2.26) tandis que les descripteurs binaires se basent sur une distance de Hamming (équation 2.27) dont

```
array([  2.,  101.,  119.,   17.,    5.,    1.,    0.,    0.,   61.,
        68.,   38.,   14.,    3.,    0.,    1.,    8.,   25.,   13.,
         9.,    5.,    2.,    3.,   12.,   14.,    8.,    4.,    3.,
         3.,    2.,    2.,    4.,    4.,   30.,  137.,  143.,   12.,
        45.,   49.,    0.,    1.,  143.,  143.,   36.,    0.,    1.,
         1.,    1.,   12.,   70.,   12.,    6.,    3.,    2.,    8.,
         7.,   18.,    7.,    1.,    2.,    1.,    3.,   10.,    6.,
        11.,   60.,   15.,    5.,    2.,  125.,  143.,   30.,   51.,
       143.,   31.,    2.,    0.,    1.,    7.,   12.,  118.,   85.,
        22.,    8.,    4.,    2.,    2.,    1.,    8.,   17.,    7.,
         1.,    1.,    4.,    8.,    4.,    8.,   18.,   38.,    4.,
         1.,   23.,   82.,   73.,   46.,   97.,    3.,    0.,    0.,
         0.,    5.,   43.,  117.,   54.,   12.,    1.,    0.,    0.,
         0.,    0.,   17.,   16.,    4.,    2.,    5.,    4.,    3.,
         3.,   15.], dtype=float32)

array([181,  55,  58,  64,  40, 201, 161, 158, 118, 123, 248, 235,  88,
        23,  30, 215,  46,  89, 187, 147, 147, 216, 247, 210, 178, 222,
       244, 247, 153, 199, 136,  44], dtype=uint8)
```

FIGURE 2.13 – Exemples de descripteurs SIFT (en haut) et BRIEF (en bas).

le calcul peut-être obtenu en un cycle d'horloge selon l'architecture de l'unité de calcul utilisée :

$$\mathrm{D}_{\mathrm{euclid}}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.26}$$

$$\mathrm{D}_{\mathrm{hamming}}(x,y) = \sum_{i=1}^{n} x_i \oplus x_j \tag{2.27}$$

2.3.1 Génèse

Bien que nous nous intéressions dans cette étude principalement aux descripteurs binaires modernes dont les travaux ont commencé dans les années 2000, il en existait auparavant d'autres qui répondaient à certains critères de définitions énoncés ci-dessus. C'est le cas des motifs binaires locaux (local binary pattern en anglais, souvent abrégé LBP) proposés par Ojala *et al.* en 1996 [Ojala et al., 1996]. La méthode de calcul des éléments du descripteur utilisée dans la version originale , illustrée par la figure 2.14 est la suivante : pour chaque pixel P étudié, les huit pixels voisins p_{vi} sont seuillés en fonction de la valeur de ce premier, suivant le test τ :

$$\tau(\mathrm{P}; p_v; i) := \begin{cases} 1 \text{ si } \mathrm{P} < p_{vi} \\ 0 \text{ sinon} \end{cases} \tag{2.28}$$

Les bits obtenus sont ensuite interprétés selon un ordre arbitraire afin de former un nombre, traditionnellement du bit le plus fort au plus faible selon un sens anti-trigonométrique en partant du voisin supérieur-gauche. Ainsi, dans l'exemple étudié le nombre obtenu 11101100 en base 2 correspond à 236 en décimal.

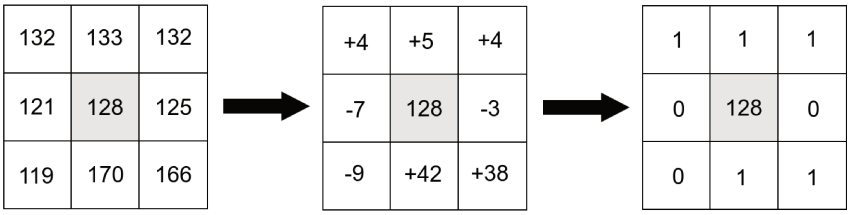


FIGURE 2.14 – Exemple de construction d'un vecteur binaire LBP : calcul de la différence du pixel central avec ses 8 voisins puis seuillage (0 ou 1) en fonction du signe de la différence.

L'utilisation classique est de découper l'image en sous-régions, qui peuvent être ou ne pas être de taille identique, de calculer un histogramme des réponses LBP pour chaque pixel dans chacune d'entre-elles et de se servir de la concaténation des histogrammes comme descripteur global de l'image. Dans le cas où les régions sont de tailles différentes, les n histogrammes $\mathrm{H}_{1,\ldots,n}$ doivent bien entendu être normalisés :

$$\mathrm{N}_i = \frac{\mathrm{H}_i}{\sum_{j=1}^{n} \mathrm{H}_j} \tag{2.29}$$

LBP étendu

La première version de cet algorithme avait recours à l'étude des huit voisins immédiats du pixel pris en compte. Une telle approche ne permet pas de prendre en charge des changements d'échelle, ce qui est l'objectif de la modification proposée par Ojala *et al.* en 2002 [Ojala et al., 2002] qui généralise l'algorithme pour prendre en compte P pixels sur un cercle de rayon R. Ceux-ci sont définis sur un cercle centré autour du pixel c, et l'emplacement du pixel p est obtenu en calculant $(-\mathrm{R}\sin(2\pi p/\mathrm{P}), \mathrm{R}\cos(2\pi p/\mathrm{P})$.

Les auteurs proposent aussi un principe de « motifs uniformes » (U). Observant que certains motifs qui correspondent à des coins ou à des bords sont plus à même de caractériser l'information visuelle de texture que d'autres, ils définissent une nouvelle façon de calculer les réponses LBP :

$$\mathrm{LBP}^{u}_{\mathrm{P,R}} = \begin{cases} \sum_{p=1}^{\mathrm{P}} 2^p s(g_p - g_c) \text{ si } \mathrm{U}(\mathrm{N}(\mathrm{P},\mathrm{R})) \leq 2 \\ \mathrm{P}+1 \text{ sinon} \end{cases} \tag{2.30}$$

avec

$$\mathrm{U}(\mathrm{P},\mathrm{R}) = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{\mathrm{P}-1} |s(g_p - g_c) - s(g_{p-1} - g_c| \tag{2.31}$$

où g_i correspond au niveau de gris du pixel i. Ce calcul permet d'obtenir le nombre de basculements de bits dans le vecteur binaire. Si celui-ci est inférieur ou égal à deux, la suite de bits à 1 peut-être vue comme continue (en prenant en compte une notion de cyclicité). Ainsi, dans le cas classique d'un mot LBP de un octet, sur les 256 vecteurs possibles 58 sont uniformes impliquant 59 mots $\mathrm{LBP}_{\mathrm{P,R}}$ possibles.

Modifications diverses

Une des richesses de cet algorithme est de pouvoir ajouter des modifications très facilement grâce à sa simplicité. Par exemple, en plus de l'utilisation des motifs uniformes

nous pouvons définir une invariance par rotation (ir) en calculant le nombre maximal possible lors d'un décalage des bits dans un cycle complet :

$$\mathrm{LBP}_{ir} = \max\{\mathrm{ROR}(\mathrm{LBP}, j) | j = 0, 1, ..., \mathrm{J}-1\} \tag{2.32}$$

Cette stratégie réduit toutefois encore plus le nombre de possibilités différentes de mots LBP, passant ainsi de 256 valeurs possibles à 36 seulement lors du codage sur un octet.

A titre d'exemple, notons une façon très simple d'assurer une résistance au bruit en appliquant sur chaque pixel p de la version étendue un lissage gaussien dont l'écart-type est fonction de la longueur R du rayon d'étude.

Nous nous intéressons maintenant aux descripteurs binaires modernes qui ont vu le jour à la fin des années 2000. Nous faisons le choix de séparer ces derniers en deux catégories que nous appelons respectivement première et seconde génération. Les descripteurs de première génération sont des dérivés de BRIEF où le calcul du vecteur caractéristique est fait à partir de différences d'intensités lumineuses et ceux de deuxième génération ont des approches plus complexes.

2.3.2 Première génération

En dehors de leur pouvoir discriminant avéré, le succès des descripteurs dérivés des motifs binaires locaux peut trouver son origine dans plusieurs facteurs. Pour reprendre la devise de Mikhaïl Kalachnikov, « Quelque chose de complexe n'est pas utile et tout ce qui est utile est simple ». Or, ces descripteurs sont d'une redoutable simplicité : en plus de la rapidité d'implantation, il n'est pas nécessaire d'avoir des connaissances particulières en traitement d'image, signal ou mathématiques pour les programmer, ce qui a facilité la diffusion dans les milieux académiques et industriels. De plus, le contexte applicatif d'origine étant la classification de texture dans l'analyse d'images de visages comme la vidéo surveillance, ils se sont retrouvés particulièrement adaptés à la problématique de traitement en temps réel, comme un flux vidéo, étant donné la rapidité de calcul du vecteur caractéristique, basé sur des calculs de différences.

BRIEF : Binary Robust Independant Element Features (2010)

A partir d'une image convertie en niveaux de gris, dans une sous-image autour du point d'intérêt étudié, l'algorithme [Calonder et al., 2010] dispose des paires de pixels et attribue pour chacune d'entre elle un bit, résultat de l'évaluation d'un test τ de calcul de différence d'intensité lumineuse :

$$\tau(p; x, y) := \begin{cases} 1 \text{ si } p(x) < p(y) \\ 0 \text{ sinon} \end{cases} \tag{2.33}$$

Ainsi, chaque paire constitue un élément du vecteur caractéristique et leur nombre pouvant être un paramètre de l'application, l'augmenter améliore le pouvoir discriminant mais réduit le temps de calcul ainsi que la consommation mémoire et le temps nécessaire pour réaliser l'appariement. L'auteur propose entre 128 et 512 paires mais c'est généralement 256 qui est utilisé, réalisant un bon compromis entre rapidité et efficacité.

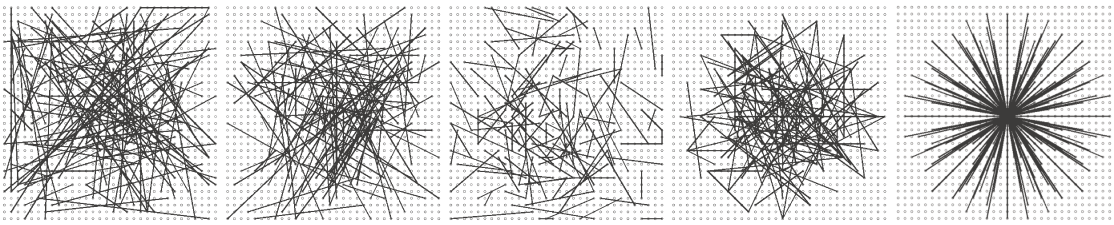


FIGURE 2.15 – Les 5 motifs d'échantillonnage de paires proposés par BRIEF. Le premier (en partant de la gauche), basé sur l'aléatoire, est celui retenu dans les implantations classiques. [Calonder et al., 2010]

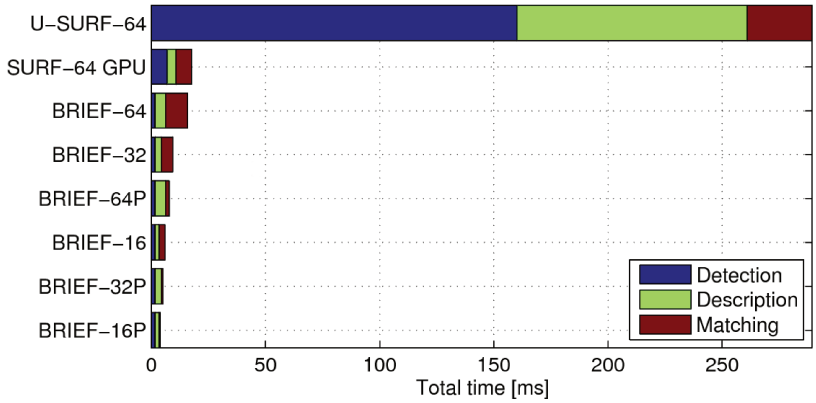


FIGURE 2.16 – Comparaison des temps de calculs des différentes versions de BRIEF avec SURF. La version CPU la plus lourde reste toujours en dessous de l'exécution sur GPU de SURF. [Calonder et al., 2010]

L'auteur de l'article propose par ailleurs plusieurs manières de distribuer les paires de pixels P(x)/P(y) en fonction de plusieurs hypothèses réalisées, illustrées par la figure 2.15, de gauche à droite :

1. aléatoire complet : l'information serait plus efficacement couverte si les paires sont réparties équitablement dans la sous-image, sans tentative d'influencer la distribution,
2. l'information serait plus intéressante autour du point d'intérêt : les paires sont toujours tirées au hasard mais suivant une distribution gaussienne centrée sur celui-ci,
3. partant de la même idée mais supposant que les paires courtes sont plus chargées en information, x est tiré selon une distribution gaussienne autour du centre tandis que y est lui obtenu selon une distribution gaussienne centrée autour de x,
4. similaire à la deuxième hypothèse mais tentant d'ajouter plus d'ordre : x et y sont tirés aléatoirement sur une grille polaire,
5. se même, $\forall i$, $x_i = (0,0)$ et y_i est tiré sur une grille polaire.

En conclusion, les résultats de ses expérimentations indiquent que la première hypothèse (l'aléatoire) donne les meilleurs résultats et que la cinquième est de loin la dernière. Il est enfin important de noter que l'information étant récupérée directement au niveau du pixel lui même, cet algorithme est très sensible aux inévitables problèmes de bruit. Afin d'y remédier, la fenêtre d'étude est lissée à l'aide d'un flou gaussien. Le bénéfice de robustesse au bruit étant considéré comme supérieur à la perte d'information induite.

Toutefois, bien qu'étant une avancée innovante, cet algorithme possède des défauts majeurs : il ne propose aucun mécanisme efficace de résistance au changement d'échelle et de rotation mis à part de calculer un ensemble de réponses pour chaque sous-image en appliquant des transformations de perspective, ce qui alourdit considérablement le temps de calcul et l'espace mémoire. De surcroit, son efficacité étant maintenant limitée par rapport aux contributions récentes. Ne reste que sa simplicité et son temps de calcul très faible (figure 2.16).

Color BRIEF

Cette contribution [Kottman, 2011] part d'un problème classique en vision par ordinateur : la gestion de la couleur. La plupart du temps, cette information est perdue en convertissant l'image en niveaux de gris et le problème n'est pas traité. L'auteur propose une modification de BRIEF où à chaque paire est attribué un canal de couleur (RGB). Très simplement, le test τ devient alors :

$$\tau(p; x, y, c) := \begin{cases} 1 \text{ si } p_c(x) < p_c(y) \\ 0 \text{ sinon} \end{cases} \tag{2.34}$$

où $c \in \{R, G, B\}$.

L'auteur de l'article avance un pouvoir discriminant plus fort, toutefois cela reste à nuancer puisque cet algorithme n'est en pratique jamais utilisé. Nous pouvons imaginer que les performances varient beaucoup selon la couleur attribuée à une paire. Or, il n'en est fait aucune mention dans la contribution.

Steered BRIEF

Modification censée apporter de la résistance au changement d'orientation [Rublee et al., 2011], Steered BRIEF propose de se baser sur la définition de l'orientation du centroïd du moment proposé par Rosen, comme étudié précédemment. A partir de l'angle d'orientation θ obtenu, il est possible de construire une matrice de rotation R_θ. Celle-ci peut-être appliquée à la matrice des coordonnées S des n paires composant le motif de calcul du descripteur.

$$S = \begin{pmatrix} x_1, ..., x_n \\ y_1, ..., y_n \end{pmatrix} \tag{2.35}$$

Nous obtenons ainsi la matrice des coordonnées orientées :

$$S_\theta = R_\theta S \tag{2.36}$$

Afin de conserver la rapidité de calcul de BRIEF, les matrices S_θ sont pré calculées et stockées dans une table de correspondance en itérant sur θ par pas de $\frac{2\pi}{30}$.

Bien que répondant à la problématique, cette modification de BRIEF n'est plus utilisée de nos jours. En effet, comme étudié par Rublee *et al.* lors de l'élaboration de ORB, les motifs BRIEF présentent de fortes corrélations dans les paires utilisées et le pouvoir discriminant du descripteur est affaibli lorsque celui-ci est orienté. L'hypothèse émise lors de cette étude était que BRIEF repose sur une orientation aléatoire des points d'intérêt, ce qui expliquerait que l'auteur original ne se soit pas plus penché sur un mécanisme de compensation au changement d'orientation.

ORB : Oriented FAST and Rotated BRIEF (2011)

Proposé par Calonder *et al.* en 2011 [Rublee et al., 2011], ORB reprend le même principe que son ainé BRIEF avec l'objectif de combler ses lacunes et d'améliorer la distribution des paires. Observant que Steered BRIEF perd en pouvoir discriminant à cause de son motif d'échantillonnage des paires, les auteurs proposent une méthode d'apprentissage (voir algorithme 2) dans le but de sélectionner un sous-ensemble de tests plus performant.

Algorithme 2 : ORB : apprentissage des paires

Données : Seuil : seuil de corrélation maximum
Résultat : R : ensemble des paires résultat
Calculer chaque test sur les sous-images d'entrainement.
T ← Trier les tests en fonction de leur distance à une moyenne de 0.5
Ajouter le premier test de T dans R
tant que *R est composé de moins de 256 éléments* **faire**
  si *il reste des tests dans T* **alors**
    T_{curr} ← test suivant dans T
  fin
  sinon
    augmenter $Seuil$
    Revenir au début de T
  fin
  $corr$ ← correlation de T_{curr} aux tests dans R
  si $corr < Seuil$ **alors**
    ajouter T_{curr} dans R
    Enlever T_{curr} de T
  fin
fin
retourner R

L'objectif est de maximiser la variance des paires tout en minimisant leur corrélation. L'algorithme est appliqué sur un ensemble de 300000 points d'intérêts extraits du jeu de données PASCAL 2006 [Everingham et al.,] où chaque point est analysé sur une fenêtre de 31×31 pixels en parcourant l'ensemble des tests possibles. Ceux-ci sont constitués d'une paire de sous-fenêtres de taille 5×5. Les tests redondants étant éliminés, 205590 tests sont conservés au total. Le motif résultant de l'apprentissage, appelé rBRIEF par les auteurs, est composé de 256 paires et est montré avec la figure 2.17 où l'on constate que celles-ci sont réparties de façon équitable.

Finalement, ORB améliore grandement les capacités discriminantes de BRIEF en apportant un mécanisme de résistance à l'orientation similaire à Steered BRIEF. La force de cette contribution est d'apporter cette modification tout en en conservant un temps calculatoire très faible, à l'inverse de descripteurs binaires plus modernes. Cela fait qu'il est encore très utilisé aujourd'hui dans des applications car il représente un bon compromis. Son seul défaut potentiellement très contraignant est toutefois l'absence d'un mécanisme de résistance au changement d'échelle.

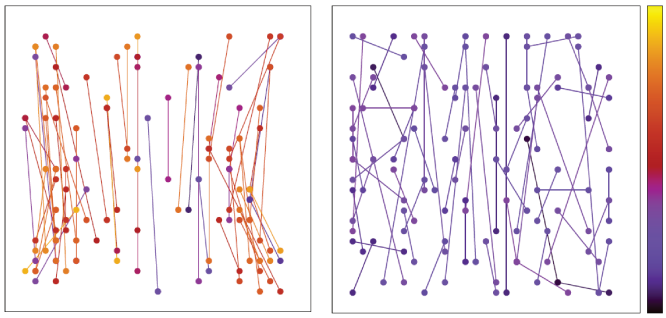


FIGURE 2.17 – Sous-ensemble de paires générées en considérant la variance des vecteurs uniquement (à gauche) et en appliquant l'algorithme 2 (à droite). Une couleur bleue indiquant un faible niveau de corrélation avec les autres paires. [Rublee et al., 2011]

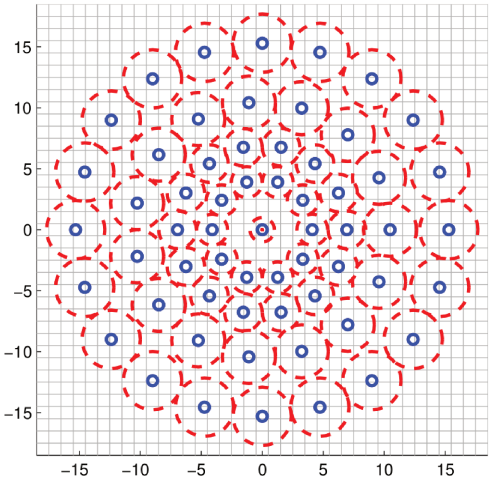


FIGURE 2.18 – Motif d'échantillonnage de BRISK. Les cercles bleus correspondent aux emplacements des points de formation des paires. Les cercles rouges correspondent à la taille des déviations standards des noyaux gaussiens utilisés pour lisser les valeurs des pixels. [Leutenegger et al., 2011]

BRISK : Binary Robust Invariant Scalable Keypoints (2011)

Comme ORB, BRISK [Leutenegger et al., 2011] entend combler les lacunes de BRIEF en s'inspirant de celui-ci. Il propose ainsi un mécanisme de résistance au changement d'orientation et en plus par rapport à ORB, au changement d'échelle. Pour cela, et au contraire de ce dernier, BRISK part sur une direction différente en reprenant le postulat abandonné par les auteurs de BRIEF sur la distribution des paires qui bénéficierait d'un motif géométrique plus ou moins imposé (figure 2.15, dernier motif). Ici, celui-ci a été dessiné à la main (figure 2.18) et sa taille diffère en fonction du paramètre d'échelle t retourné par le détecteur de point d'intérêt, lui assurant ainsi l'invariance par changement d'échelle.

Comme nous pouvons le constater sur la figure 2.18, le motif d'échantillonnage des paires est composé de 60 points p ; de ce fait l'ensemble $\mathscr{A}$ des paires possibles est d'une taille de 1770 éléments. Fort heureusement, le vecteur caractéristique n'est pas d'une taille de 1770 bits car l'algorithme n'utilise qu'une partie des paires pour le construire. A partir de $\mathscr{A}$, nous pouvons construire deux sous-ensembles ; celui des paires courtes ($\mathscr{C}$) et celui des paires longues ($\mathscr{L}$) (celles qui ne rentrent pas dans une des deux catégories

sont éliminées) :

$$\begin{aligned}\mathscr{C} &:= \{(p_i, p_j) \in \mathrm{A} \mid\mid p_j - p_i \mid\mid < \delta_{\max}\} \\ \mathscr{L} &:= \{(p_i, p_j) \in \mathrm{A} \mid\mid p_j - p_i \mid\mid > \delta_{\min}\}\end{aligned} \tag{2.37}$$

Les seuils $\delta_{\min}$ et $\delta_{\max}$ sont fixés respectivement à $13.67t$ et $9.75t$, le fait qu'ils soient fonction de l'échelle du point d'intérêt faisant partie du mécanisme de résistance au changement d'échelle.

A l'aide des L paires de l'ensemble $\mathscr{L}$, il est possible de calculer l'orientation g de la sous-image comme étant la somme des gradients locaux $g(p_i, p_j)$. L'auteur explique le choix de ne pas utiliser les paires courtes dans le calcul de l'orientation après avoir observé que leurs gradients locaux avaient tendance à s'annuler les uns les autres, seules les paires longues étaient collectivement porteuses de cette information :

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{\mathrm{L}} \sum_{(p_i, p_j) \in \mathscr{L}} g(p_i, p_j) \tag{2.38}$$

où le gradient local $g(p_i, p_j)$ est défini par :

$$g(p_i, p_j) = (p_j - p_i) \frac{\mathrm{I}(p_j, \sigma_j) - \mathrm{I}(p_i, \sigma_i)}{\mid\mid p_j - p_i \mid\mid^2} \tag{2.39}$$

avec $\mathrm{I}(p_i, \sigma_j)$ l'intensité lumineuse du pixel p_i après application d'un filtre gaussien d'écart-type σ_i. Ainsi, le motif d'échantillonnage est tourné d'un angle $\alpha = \arctan2(g_y, g_x)$ autour du point d'intérêt étudié.

Les 512 paires courtes de l'ensemble $\mathscr{C}$ sont utilisées dans la construction du vecteur caractéristique suivant un test τ similaire aux descripteurs habituels :

$$\tau(p; i; j; \alpha; \sigma) := \begin{cases} 1 \text{ si } \mathrm{I}(p_j^\alpha, \sigma_j) < \mathrm{I}(p_i^\alpha, \sigma_i) \\ 0 \text{ sinon} \end{cases} \tag{2.40}$$

L'auteur avance trois intérêts à l'utilisation de ce motif d'échantillonnage : l'application de filtres gaussiens locaux avec leur propre déviation standard en fonction de l'emplacement du point permet de réduire la perte d'information en évitant de filtrer deux points p d'une paire trop proches. De plus, les comparaisons étant restreintes spatialement, les variations d'intensité lumineuse n'ont besoin d'être consistantes que localement. Enfin, mais plus discutable, puisque le nombre de points est bien plus faible que le nombre de paires (60 contre 1770), cela réduirait la complexité d'avoir à calculer les intensités lumineuses des points. Cependant, le mécanisme de calcul d'orientation de la sous-image alourdit de façon non négligeable le temps de traitement de cet algorithme.

FREAK : Fast Retina Keypoint (2012)

Avec FREAK, Alahi *et al.* poursuivent la voie ouverte par BRISK en faisant reposer leur algorithme sur un motif dessiné à la main (figure 2.19). Celui utilisé ici ressemble fortement au motif de BRISK mais il se réclame toutefois comme étant « bio-inspiré » ; ses caractéristiques sont inspirées d'une observation du vivant en partant du principe que l'évolution

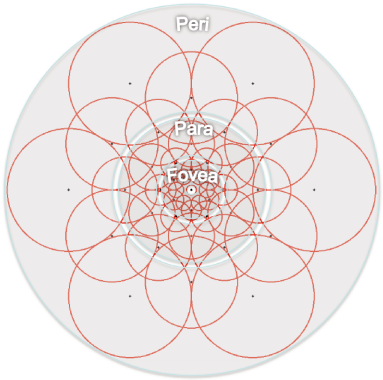


FIGURE 2.19 – Motif d'échantillonnage des paires de FREAK. La distribution des points et la taille des noyaux gaussiens associés suit la répartition des cellules photo réceptrices de la rétine. [Alahi et al., 2012]

aura sélectionné un système performant, même si ce point est discutable puisqu'en pratique nous observons souvent dans la nature un compromis. Les points servant à la formation des paires sont distribués de façon à approcher la répartition des cellules réceptives de la rétine. Il en va de même pour la taille des déviations standards des noyaux gaussiens utilisés qui varie exponentiellement en s'approchant des bords. Autre changement notable par rapport à BRISK, les zones couvertes par les noyaux gaussiens se superposent, permettant d'ajouter de l'information redondante qui améliore le pouvoir discriminant.

De façon similaire à ORB par rapport à BRIEF, FREAK s'attache à améliorer la sélection des paires par rapport à sa source d'inspiration. BRISK est en effet relativement simple sur cet aspect, ne reposant la constitution des paires que sur un calcul de distances spatiales. Or, comme ORB le constatait pour BRIEF, cela ne garantit pas l'absence de corrélation. Ainsi, les auteurs utilisent un algorithme similaire pour apprendre un ensemble de paires plus discriminantes en maximisant la variance tout en minimisant la corrélation. Le résultat, un ensemble de 512 paires, est montré avec la figure 2.20.

Toujours dans l'optique d'une approche bio-inspirée, les auteurs regroupent les paires selon un critère « grossier à fin » (*coarse to fine*) en quatre groupes de 128 paires chacun, le premier concerne principalement les paires en bordure du motif et le dernier au centre de celui-ci.

Cette approche permet aux auteurs d'améliorer la rapidité d'appariement des points d'intérêt : seul les 128 premiers bits « périphériques » des vecteurs sont utilisés lors de la première phase, éliminant les candidats en dessous d'un certain seuil.

Le calcul de l'orientation est similaire à celui utilisé pour BRISK, par estimation du gradient local à l'aide des paires du motif. Celles-ci ne sont toutefois qu'au nombre de 45, formant un motif symétrique centré sur le point d'intérêt.

ALOHA : Aggregated LOcal HAar (2012)

Cet algorithme [Saha and Démoulin, 2012] s'éloigne légèrement des descripteurs binaires étudiés jusqu'à présent en s'inspirant à la fois de BRIEF et de la méthode de Viola-Jones en reconnaissance d'objet [Viola and Jones, 2001]. L'idée principale est que l'information au niveau du pixel uniquement n'est pas la meilleure source utilisable ; les auteurs

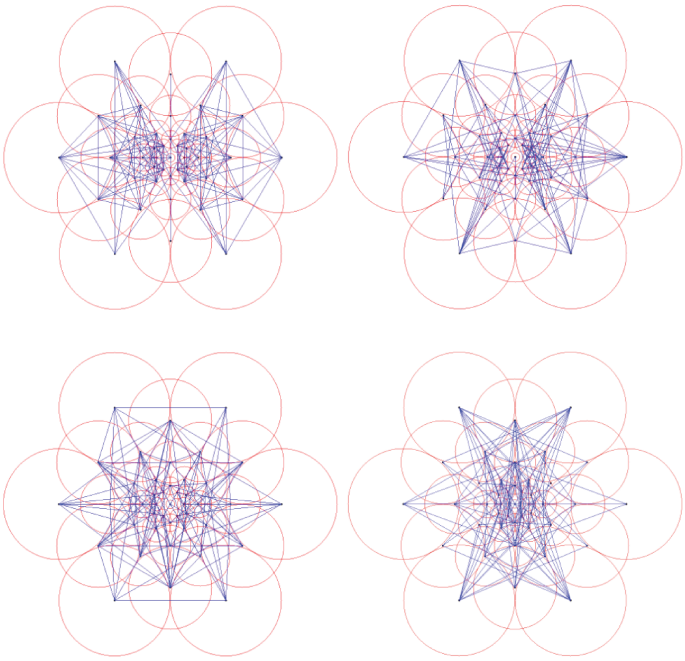


FIGURE 2.20 – Regroupement des paires apprises en quatre groupes suivant une approche « grossier à fin ». [Alahi et al., 2012]

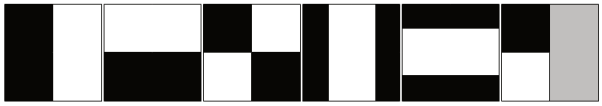


FIGURE 2.21 – Exemples de décompositions de surfaces où sont calculées les ondelettes de Haar avec ALOHA : seules les surfaces en noir ou en blanc sont utilisées, elles sont de tailles identiques.

proposent ainsi de recourir à des ondelettes de Haar en faisant reposer le test τ sur un calcul de différence de moyenne d'intensité de deux surfaces de pixels :

$$\tau(P;X,Y) := \begin{cases} 1 \text{ si } \overline{P_X} > \overline{P_Y} \\ 0 \text{ sinon} \end{cases} \tag{2.41}$$

où $\overline{P_X}$ et $\overline{P_Y}$ représentent les intensités moyennes de deux surfaces X et Y dans la sous-image P.

Le système classique d'échantillonnage à l'aide de paires est alors remplacé par un ensemble de 32 divisions de la sous-image en 2 groupes de pixels de tailles identiques, un extrait de configurations étant présenté avec la figure 2.21. Sur une surface P, l'algorithme réalise L tests τ à l'aide des L premiers motifs afin de construire un vecteur binaire. Cette surface P est ensuite divisée en 4 sous-surfaces et chacune d'elles est ensuite partitionnée à nouveau, récursivement, où le test τ est à nouveau appliqué. Ce processus de décomposition est montré à l'aide de la figure 2.22. Sur les surfaces de niveau 0 et 1 (P et P_1), τ est appliqué 32 fois et 6 fois seulement sur celles de niveau 2 (P_2). Le vecteur binaire est ainsi composé de $(1+4) \times 32 + 16x6 = 256$ bits.

Étant donné que l'algorithme utilise des images intégrales, le calcul d'une somme d'une région se fait au moyen d'une addition et 2 soustractions seulement ; un vecteur est donc obtenu avec $2x3x256 = 1536$ opérations élémentaires seulement.

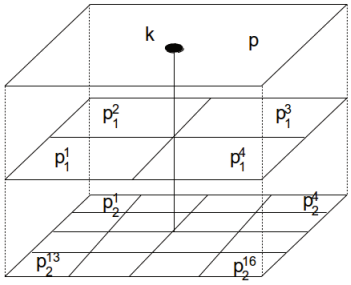


FIGURE 2.22 – Décomposition récursive par ALOHA d'une sous-fenêtre en plusieurs surfaces où sont appliquées les ondelettes de Haar ; 32 sur chacun des 2 premiers niveaux et 16 pour celles du troisième et dernier niveau.

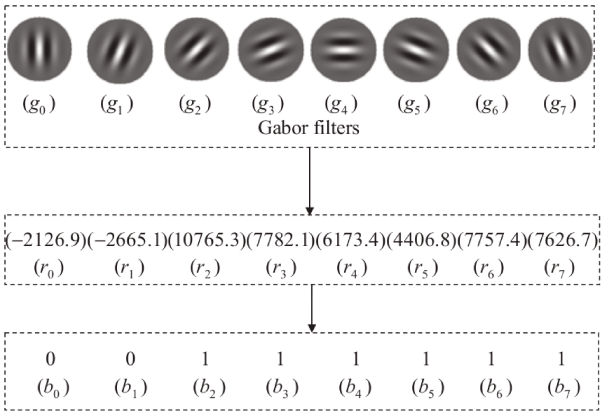


FIGURE 2.23 – La construction d'un vecteur binaire avec BGP obtenue par étude du signe de la somme des réponses d'un filtre de Gabor. Les filtres partagent les mêmes paramètres si ce n'est celui de l'orientation, 8 possibles par changement de $\pi/4$. [Zhang et al., 2012]

En revanche, malgré l'originalité de son approche parmi ses concurrents, ALOHA ne propose pas de mécanisme de résistance au changement d'échelle ou à l'orientation en plus de ne pas être libre de droits, ce qui explique en partie pourquoi il est peu utilisé aujourd'hui.

BGP : Binary Gabor Pattern (2012)

Principalement conçu pour de la classification de texture, BGP s'éloigne un peu des descripteurs binaires présentés précédemment [Zhang et al., 2012]. Il repose notamment sur les filtres de Gabor [Gabor, 1946] et n'est pas basé sur un principe de paires de pixels. Sur le point étudié, l'algorithme applique J filtres de Gabor partageant les mêmes caractéristiques à l'exception de l'orientation. Les pixels résultants sont additionnés pour former une réponse, et, à l'instar des LBP, le bit résultat est fonction du signe de celle-ci. Le vecteur est donc défini comme étant :

$$\text{BGP} = \sum_{j=0}^{\text{J}} b_j \times 2^j \tag{2.42}$$

où b_j correspond au bit de la j_{eme} réponse ; ce processus est illustré avec la figure 2.23. Notons au passage que les bits de poids forts sont ceux situés sur la droite.

Les auteurs proposent de traiter le problème de la rotation en utilisant le même procédé employé chez certaines variantes de LBP, comme expliqué précédemment : les bits sont

décalés cycliquement vers la droite j fois et la valeur maximum est retenue comme le vecteur invariant à la rotation, BGP_{ri} :

$$\mathrm{BGP}_{ri} = max\left\{\mathrm{ROR}(\mathrm{BGP}, j) | j = 0, 1, ..., \mathrm{J}-1\right\} \tag{2.43}$$

La conséquence immédiate de l'utilisation de cette astuce est de limiter sévèrement le nombre de réponses différentes possibles. Ainsi, pour un octet, il existe 36 valeurs différentes de BGP_{ri}.

Tout comme LBP, les réponses sont utilisées pour former un histogramme dans un but de classification de texture. Ainsi, le vecteur descripteur final n'est pas binaire.

LATCH : Learned Arrangements of Three Patch Codes (2015)

Dernier né avec BOLD (cf descripteur suivant) des descripteurs de première génération (i.e. basés sur un calcul de différences), LATCH [Levi and Hassner, 2016] tente de combler ce que les auteurs estiment être une stratégie par défaut et perfectible des autres descripteurs : la gestion du bruit. Puisque l'information est récupérée directement dans l'intensité lumineuse du pixel, une modification brutale liée au bruit peut grandement impacter le résultat. C'est pour cela que la fenêtre d'étude est généralement lissée à l'aide d'un flou gaussien. Ici, les auteurs proposent de se baser non sur la valeur d'un pixel mais sur celle de régions de pixels : l'information visuelle étant spatialement étendue, c'est donc une alternative au lissage gaussien habituel.

L'ensemble de paires de pixels est remplacé par un ensemble $\hat{\mathrm{S}}$ de T triplets de régions, illustré par la figure 2.24 : une région « d'ancrage » $\mathrm{P}_{t,a}$ et deux régions « compagnons » $\mathrm{P}_{t,1}$ et $\mathrm{P}_{t,2}$. Le test τ est alors un test de similarité de la région d'ancrage aux deux autres régions compagnons en calculant leur norme de Frobenius, ou Hilbert-Schmidt :

$$\tau(\mathrm{W}; \hat{\mathrm{S}}_t) := \begin{cases} 1 \text{ si } ||\mathrm{P}_{t,a} - \mathrm{P}_{t,1}||_{\mathrm{F}}^2 > ||\mathrm{P}_{t,a} - \mathrm{P}_{t,2}||_{\mathrm{F}}^2 \\ 0 \text{ sinon} \end{cases} \tag{2.44}$$

où $||.||_{\mathrm{F}}^2$ correspond à la norme de Frobenius défini comme la norme euclidienne du vecteur des valeurs singulières de la matrice constituée des pixels de la région P_t étudiée.

A nouveau, se pose la question de la distribution des éléments d'échantillonnage (ici, les triplets). Comme nous avons vu jusqu'à présent, il est possible de construire un ensemble basé sur le hasard, d'utiliser un motif réalisé à la main s'inspirant d'une forme géométrique particulière ou d'avoir recours à un apprentissage. C'est cette dernière méthode qui a été retenue pour cet algorithme.
À partir d'une base de données d'images contenant plusieurs vues sous différentes conditions, 400000 sous-fenêtres sont extraites à partir de points d'intérêts obtenus avec le détecteur de coin de Harris. À l'aide de méthodes de reconstruction de scènes par stéréo, des paires de points (500000) sont labellisées comme étant « identiques » ou « différentes », ces deux ensembles sont de tailles identiques. Les auteurs ne précisent pas comment aussi nous prenons l'hypothèse qu'ils forcent la construction dans ce but, de façon purement arbitraire. Sont ensuite construits 56000 triplets par sélection aléatoire du pixel de la région d'ancrage $\mathrm{P}_{t,a}$ ainsi que les coordonnées des deux compagnons $\mathrm{P}_{t,1}$ et $\mathrm{P}_{t,2}$. L'évaluation d'un arrangement se fait ensuite en construisant la somme du nombre de fois où celui-ci retourne le bit correct pour le label « identique » ou « différent ». De plus, comme

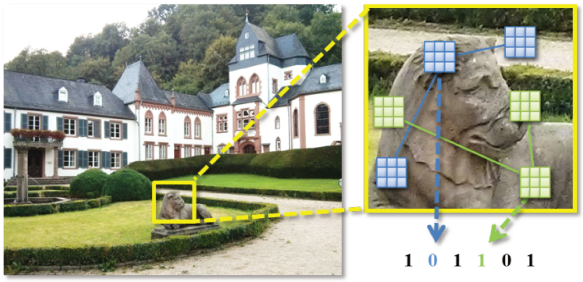


FIGURE 2.24 – Illustration du procédé de calcul du test τ de LATCH pour aboutir à un bit au moyen d'un triplet de surfaces. [Levi and Hassner, 2016]

l'ont fait ORB et FREAK, dans le but d'éviter la situation où les arrangements sont fortement corrélés, ceux-ci sont ajoutés de façon itérative en étudiant la corrélation du nouvel ajout dans l'ensemble en cours de construction ; un triplet est valide si la corrélation absolue est plus petite qu'un seuil γ ($\gamma = 0.2$ dans l'article original). Par simple mimétisme par rapport aux autres descripteurs, c'est une taille de 256 triplets (donc des vecteurs de 256 bits) qui a d'abord été choisie puis finalement conservée après expérimentations et observation du bénéfice.

Les auteurs ont aussi étudié les différentes tailles possibles des régions centrées autour des P_t. Ils remarquent que les résultats d'appariements sont toujours améliorés en agrandissant celles-ci, au détriment du temps de calcul. De surcroit, étant donné que le gain de mise en correspondance s'atténue grandement à partir d'une taille 9×9, les auteurs recommandent d'utiliser 7×7 qui semble être le meilleur compromis.

Cela nous amène à l'observation finale. Avec une taille fixe des régions d'études et de construction des triplets, cet algorithme ne propose pas réellement de mécanismes de résistance au changement d'échelle ni de compensation au changement d'orientation. Enfin, par son calcul de norme de Frobenius sur 3 surfaces pour obtenir un bit, il est sensiblement plus long que nombre de descripteurs binaires (à titre d'exemple, approximativement trois fois plus long que BRIEF). Son temps de calcul reste malgré tout bien en dessous des descripteurs flottants et ses performances sont annoncées comme excellentes, ce qui lui a permis d'être intégré dans la bibliothèque OpenCV lors du passage de sa version 3.0. Il est donc fortement probable qu'une amélioration de cet algorithme arrive prochainement pour corriger ses lacunes.

BOLD : Binary Online Learned Descriptor (2015)

Ce descripteur [Balntas et al., 2015], particulièrement astucieux, est le résultat d'une analyse très poussée de l'état de l'art sur ce sujet en prenant en compte toutes les innovations récentes, y compris celles des algorithmes de « deuxième génération » que nous allons étudier juste après. Nous plaçons malgré tout cette contribution parmi les premières générations (même si la frontière est un peu floue) puisque les tests sont des simples calculs de différences d'intensités.

La base de cette contribution repose sur la constatation que les dimensions d'un descripteur binaire ne sont pas toutes autant porteuses d'information, et pire encore, qu'il n'existe pas de configuration globale optimale comme illustré par la figure 2.25. L'idée est

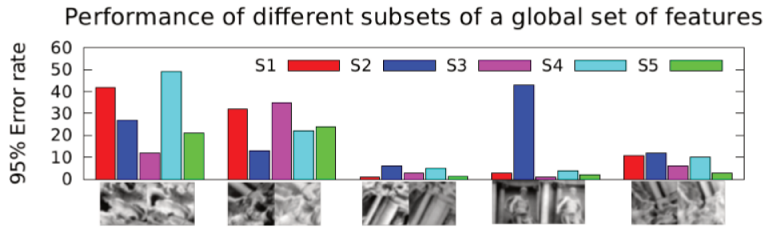


FIGURE 2.25 – Taux d'erreurs lors de la mise en correspondance de descripteurs selon différents ensembles de configurations d'échantillonnage de tests binaires et de sous-régions. [Balntas et al., 2015]

donc d'adapter la distribution des paires selon la configuration locale de la sous-fenêtre étudiée par le point d'intérêt. Là où les méthodes comme ORB cherchent à apprendre la meilleure configuration globale, BOLD propose de déterminer la meilleure configuration locale par un processus en ligne en suivant une astucieuse analyse de l'entropie d'un ensemble d'échantillonnages qui aura été appris auparavant.

Les résultats sont revendiqués comme étant supérieurs aux descripteurs flottants. Le temps de calcul est très légèrement plus long que les autres descripteurs binaires de première génération en raison de la sélection en ligne du meilleur échantillonnage mais reste considérablement plus faible que les méthodes reposant sur des calculs de gradient, ce qui fait de ce descripteur moderne une contribution innovante dans le domaine.

2.3.3 Seconde génération

Comme nous avons vu en présentant les descripteurs binaires de première génération qui sont basés sur BRIEF, ceux-ci ont chronologiquement et globalement porté des innovations à la complexité croissante. Ainsi, puisque au fur et à mesure que les « bonnes idées » ont été découvertes, c'est tout naturellement que d'autres pistes ont été travaillées en parallèle. Les descripteurs que nous présentons maintenant ne sont pas basés sur des modifications de BRIEF même si les noms sont parfois trompeurs, comme c'est le cas pour D-BRIEF.

D-BRIEF : Discriminative BRIEF (2012)

Cette méthode [Trzcinski and Lepetit, 2012] réalise la construction du vecteur caractéristique par application d'un ensemble de projections w_i sur la région d'étude x dont les résultats sont seuillés avec des paramètres τ_i. Autrement dit :

$$\forall_{i \in 1,\ldots,N} b_i = \text{signe}(w_i^T x + \tau_i) \tag{2.45}$$

L'originalité de cette approche réside, entre autres dans un souci de performance, dans la conception des projections w_i comme étant des combinaisons linéaires de primitives graphiques contenues dans un dictionnaire D comme illustré par la figure 2.26. Ainsi, $w_i = Ds_i$ avec s un vecteur binaire de coefficients. Nous avons donc le calcul d'une projection comme étant défini par :

$$w_i^T = (Ds_i)^T = \sum_{\forall j | s_{ij} \neq 0} s_{ij} D_j^T \tag{2.46}$$

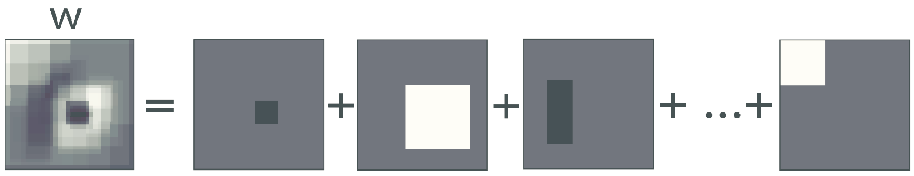


FIGURE 2.26 – D-BRIEF : décomposition d'une projection en combinaison linéaire de primitives graphiques. [Trzcinski and Lepetit, 2012]

Encore une fois, le recours aux images intégrales permet de réaliser les opérations $D_j^T x$ en une poignée d'instructions élémentaires. Toutefois nous notons que des détails importants ne sont pas abordés comme l'unicité des projections ou le choix des primitives graphiques.

Les auteurs proposent trois types de dictionnaires mais leurs expérimentations n'en ayant retenu qu'un seul, nous détaillons uniquement celui qui est utilisé pour l'exemple de décomposition de la figure 2.26. Appelé RECT pour *Rectangular filters*, il est réalisé en générant un ensemble de filtres rectangulaires de différentes tailles à des emplacements différents, en considérant uniquement ceux dont la hauteur ou la largeur peut s'écrire sous la forme $3p+1$ avec p un entier naturel. C'est le plus volumineux des dictionnaires étudiés avec une taille de 34596 éléments contre 1024 pour les deux autres qui sont respectivement basés sur des filtres gaussiens ou box.

Bien entendu, la taille et le caractère abstrait des ensembles $\{D, s_i, \tau_i\}$ empêchent une exploration manuelle de ces paramètres. Aussi, la stratégie employée ici rappelle celle utilisée par ORB en minimisant la distance de Hamming entre deux descripteurs de points d'intérêt similaires et à l'inverse la maximiser lorsque ceux-ci sont différents. C'est un problème de minimisation où les vecteurs s_i doivent être parcimonieux.

BinBoost (2012)

BinBoost [T. Trzcinski and Fua, 2012] est un peu particulier en cela qu'il s'agit de l'application des méthodes de *boosting* [Schapire, 1990] à des ensembles de descripteurs binaires. L'objectif est donc, dans une région d'étude x, de trouver un descripteur $C(x) = [C_1(x), ...C_D(x)]$ pour construire un vecteur binaire de D bits. En se basant sur Adaboost, nous avons :

$$C_d(x) = \text{signe}(b_d^T h_d(x)) \tag{2.47}$$

où $h_d(x) = [h_{d,1}(x)...h_{d,K}(x)]^T$ sont K classifieurs faibles pondérés par le vecteur b. Notons que pour idéalement coller aux conditions d'un problème de *boosting*, les valeurs prises par les bits ne sont pas les traditionnels $\{0,1\}$ mais $\{-1,+1\}$ en raison du produit réalisé pendant le *boosting*.

Les classifieurs faibles utilisés prennent en compte les orientations des gradients de l'image en intensité lumineuse et sont paramétrés par une sous-surface rectangulaire R de la surface x, une orientation e et un seuil T :

$$h(x; R, e, T) = \begin{cases} 1 \text{ si } \psi_{R,e}(x) \leq T \\ -1 \text{ sinon} \end{cases} \tag{2.48}$$

avec

$$\psi_{R,e}(x) = \sum_{m \in R} \xi_e(x,m) / \sum_{e' \in \Phi, m \in R} \xi_{e'}(x,m) \tag{2.49}$$

et

$$\xi_e(x,m) = \max(0, \cos(e - o(x,m)) \tag{2.50}$$

où $o(x,m)$ est l'orientation du gradient local dans x à l'emplacement m. Celle-ci est discrétisée pour prendre une des q valeurs dans l'ensemble $\Phi = \{0, \frac{2\pi}{q}, \frac{2\pi}{q}, ..., (q-1)\frac{2\pi}{q}\}$. Dans leurs conclusions et d'après leurs expérimentations, les auteurs recommandent de prendre les valeurs $q = 8$, K = 128 et D = 64.

La recherche des paramètres est réalisée à l'aide d'une méthode d'apprentissage automatique avec un ensemble $\{(x_n, y_n, l_n)\}$ de N correspondances où l_n prend la valeur 1 si les régions x_n et y_n sont identiques et -1 sinon en minimisant le critère de perte L :

$$\mathrm{L} = \min_{\{b_d, h_d\}} \sum_{n=1}^{\mathrm{N}} \exp\left(-\gamma l_n \sum_{d=1}^{\mathrm{D}} c_d(x_n, y_n; b_d, h_d)\right) \tag{2.51}$$

où

$$c_d(x, y, b_d, h_d) = \mathrm{signe}(b_d^{\mathrm{T}} h_d(x))\mathrm{signe}(b_d^{\mathrm{T}} h_d(y)) \tag{2.52}$$

et

$$\gamma = \nu \frac{1}{2} \log(\frac{1 + r_1}{1 - r_1}) \tag{2.53}$$

où $r_1 = \sum_{n=1}^{\mathrm{N}} \mathrm{W}_1(n) l_n c_1(x_n, y_n)$ avec $\nu = 0.4$ et W_1 une fonction de pondération comme utilisée dans Adaboost. Ainsi, minimiser le critère L revient à réduire la distance de Hamming entre deux correspondances correctes et inversement à l'augmenter dans le cas contraire.

Edge-SIFT (2013)

Ce descripteur original [Zhang et al., 2013] utilise le détecteur de SIFT pour construire son vecteur binaire à partir de l'image des contours binarisée de la région d'étude. Pour cela, les auteurs commencent par appliquer l'algorithme de détection de contours de Canny sur la région d'étude de taille D × D, ils obtiennent donc une matrice binaire contenant D^2 bits. Pour optimiser ce processus ils conseillent toutefois de normaliser l'échelle des régions étudiées. La matrice binaire résultat est le vecteur descripteur de cette méthode. Celui-ci étant très simple, l'intérêt de cette contribution réside dans la méthode d'appariement proposée.

Pour ce faire, les auteurs définissent une première mesure de similarité qui est la suivante :

$$\mathrm{Sim}(\mathrm{A}, \mathrm{B}) = 2 \times \frac{\sum_{i=1}^{\mathrm{D}^2} (a_i \times b_i)}{(\mathrm{N_A} + \mathrm{N_B})} \tag{2.54}$$

où A et B sont deux matrices binaires, N les poids binaires associés (nombre de bits à 1) et a_i, b_i les bits d'indice i. Toutefois, cette mesure naïve ne prend pas en compte l'orientation du gradient local et augmente la similarité pour des bits qui partagent le même emplacement dans A et B sans partager la même structure sous-jacente.

Ils proposent donc une version améliorée qui nécessite de découper la région d'étude en quatre sous-régions, une pour chaque quantification de l'orientation locale par pas d'angle $\pi/4$. Les pixels correspondant à un bord (donc les bits valant 1) sont ensuite transférés dans la sous-région correspondant à leur orientation. Cette nouvelle version de leur

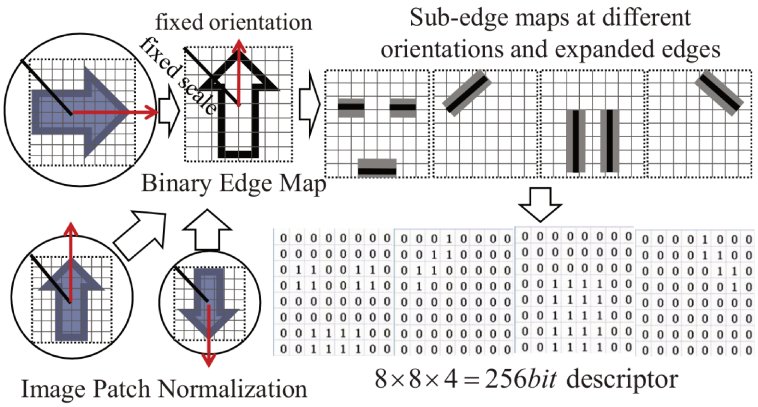


FIGURE 2.27 – Etapes de construction du vecteur caractéristique de Edge SIFT [Zhang et al., 2013]

descripteur, de taille D × D × 4 est schématisée avec la figure 2.27. Toutefois, une constatation évidente est la grande sensibilité aux transformations que peuvent subir les images de contour obtenues par Canny en cas de bruit, variation d'intensité... etc. Pour y remédier, les auteurs proposent de dilater l'épaisseur des contours extraits d'un facteur 2 × ω, ce dernier paramètre étant compris entre 1 et 2 selon la taille des régions d'études.

La nouvelle mesure de similarité devient :

$$\hat{\text{Sim}}(\text{A},\text{B}) = 2 \times \frac{\sum_{i=1}^{4\times \text{D}^2} \hat{\text{Hit}}(a_i, b_i)}{\text{N}_\text{A} + \text{N}_\text{B}} \tag{2.55}$$

où

$$\hat{\text{Hit}}(a_i, b_i) = \begin{cases} 1 \text{ si } a_i \times b_i = 1, |l_a^i - l_b^i| \leq \omega \\ 0 \text{ sinon} \end{cases} \tag{2.56}$$

Avec l l'emplacement du pixel et ω le seuil de contrôle sur la distance acceptable entre deux pixels, utilisé aussi pour la dilatation.

Notons enfin que les auteurs recommandant une valeur de D entre 16 et 32, cela représente des descripteurs de tailles respectives 1024 et 4096 ce qui est largement au dessus des méthodes concurrentes. Le poids binaire des quatre matrices étant relativement faible par rapport à leur taille, ils recommandent donc l'application de méthodes de *sparse-coding* pour compresser les vecteurs descripteurs, cela permettant de descendre aux environs de 500 bits selon la méthode et le degré de compression utilisés.

Pour conclure, notons malgré tout que l'utilisation de l'algorithme de Canny rend ce descripteur hautement sensible au floutage d'image et aux changement brutaux d'illuminations.

2.3.4 Les faux amis

Plusieurs descripteurs qui ont suivi utilisent des vecteurs binaires, cependant ils ne rentrent pas tout à fait dans le cadre de cette étude puisque ils réalisent une projection de descripteurs flottants dans un espace binaire. Sur le fond, c'est un objectif similaire à celui visé par PCA-SIFT. Cependant, l'avantage de la rapidité du temps de calcul est perdu et ne reste que celui du calcul des distances ; ils ne sont donc pas utilisables pour être générés en temps réel dans un contexte embarqué. Nous présentons malgré tout brièvement une des contributions les plus performantes.

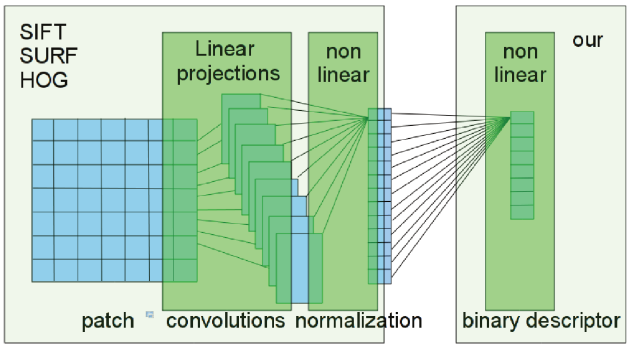


FIGURE 2.28 – Synthèse du processus de LDA-HASH. [C. Strecha and Fua, 2012]

LDA-HASH : Linear Discriminant Analysis Hash (2012)

Le but de cet algorithme [C. Strecha and Fua, 2012], résumé avec la figure 2.28, est de trouver une correspondance de l'espace des descripteurs flottants dans l'espace de Hamming à l'aide d'une transformation affine et d'une fonction de signe, qui minimise la distance de Hamming entre deux descripteurs binarisés et maximise la similarité dans l'espace de départ.

Pour reformuler le problème mathématiquement : en définissant $\mathbb{Z}$ comment étant un espace de Hamming à m dimensions $\mathbb{H}^m = \{\pm 1\}^m$, l'objectif est de trouver une fonction $y : \mathbb{R}^n \rightarrow (\mathbb{Z}, d_{\mathbb{Z}})$ dont la métrique $d_{\mathbb{Z}}$ paramètre la similarité entre deux descripteurs avec, bien entendu, $y(x)$ plus léger en consommation mémoire et $d_{\mathbb{Z}}(y(x), y(x'))$ plus rapide à calculer qu'une des normes habituellement utilisées pour les nombres réels comme la norme euclidienne. De plus, notant que bien souvent chez les descripteurs une réduction de l'espace est source de dégradations des performances, les auteurs posent des condition supplémentaire pour éviter cet écueil. Nous ne rentrerons pas plus loin dans les détails (nombreux) de l'article pour ne pas alourdir nos propos.

2.4 Conclusions

Nous avons passé en revue les contributions les plus significatives parmi les descripteurs de points d'intérêt. Souhaitant nous positionner sur le traitement nomade de l'information, nous nous sommes surtout intéressés aux contributions récentes des descripteurs binaires qui sont beaucoup plus rapides à calculer. Le tableau 2.2 résume les contributions emblématiques. Il en ressort que celles-ci, bien que très poussées dans la construction de l'information associée au point d'intérêt, ne prennent pas en compte l'inadaptabilité de ceux-ci aux images de documents : les algorithmes partent du principe que les points retournés proviennent d'emplacements originaux dans l'image ce qui n'est pas le cas pour celles représentant un document. Il en ressort alors un problème majeur au moment de réaliser l'appariement : *la confusion*. Nous discutons de ce problème dans le prochain chapitre avec la principale contribution de nos travaux pour y remédier.

Nom	Année	#bits	Commentaire
BRIEF	2010	256-512	Le premier, simple.
ORB	2011	256	Entrainement hors-ligne de la distribution des paires
BRISK	2011	512	Très performant mais sensiblement plus coûteux
ALOHA	2012	256	Non libre de droits, rarement utilisé
FREAK	2012	256	Très similaire à BRISK, bio-inspiré
D-BRIEF	2012	32	Très performant mais délicat à implanter
Edge-SIFT	2013	128	Rarement utilisé
BinBoost	2013	64	Application des techniques de boosting, très performant
LATCH	2015	256	Alternative moderne aux classiques
BOLD	2015	512	Très astucieux dans son analyse séparée des dimensions

TABLEAU 2.2 – Les principaux différents descripteurs binaires de points d'intérêt étudiés, par ordre chronologique.

Chapitre 3

La confusion

« *Le Comte - "On l'oublie trop."*
La Comtesse - "Ce ne sera pas moi."
Le Comte - "Ni moi."
Figaro, à part - "Ni moi."
Suzanne, à part - "Ni moi."
Le Comte "Il y a de l'écho ici..." »

Beaumarchais, le mariage de Figaro
Acte V scène 7

Sommaire

3.1 État de l'art

3.1.1 Définition du problème

Comme expliqué précédemment, le principe d'un vecteur caractéristique est de synthétiser l'information discriminante d'une entité. Dans le cadre de notre étude il s'agit d'un emplacement ciblé dans une image, appelé point d'intérêt. Comme sus-indiqué, cette information est une synthèse : elle n'est pas exhaustive et par analogie, cette situation peut-être ramenée au « Principe des tiroirs » qui énonce que si n objets occupent m tiroirs, et si $n > m$, alors au moins un tiroir doit contenir strictement plus d'un objet. En ce cas, il est possible pour deux points d'intérêt de partager des vecteurs caractéristiques hautement similaires, voire identiques. La confusion est alors la situation problématique qui apparait lorsque l'algorithme de mise en correspondance des vecteurs caractéristiques utilisé fonctionne correctement et réalisé l'appariement de deux points d'intérêt, correct dans l'espace des descripteurs mais faux dans la réalité spatiale.

A l'aide de la figure 3.1, Nous pouvons illustrer ce problème avec une image particulièrement sujette au problème de la confusion et un couple détecteur / descripteur fictif très simple, bi-dimensionnel.

Après application d'un algorithme (fictif lui aussi) de détection de points d'intérêts, six ont été relevés qui peuvent être divisés en deux catégories : ceux présents sur le motif de damier et ceux de la photographie. Il est important de noter que, bien que générant une situation problématique, l'algorithme qui a retourné les points de la première catégorie a très bien fonctionné selon les définitions de saillance utilisées habituellement en vision par ordinateur : ce sont des emplacements à contraste très élevé et aux angles vifs, fortement marqués, la réponse du gradient y est donc particulièrement forte. Hélas ! La présence de ce motif répétitif entraine une très grande proximité dans l'espace des descripteurs (image de gauche). Or, quand bien même ceux-ci sont conçus pour être invariant aux transformations que peut subir une image, il n'est pas possible d'avoir une invariance totale : Il existe toujours une *probabilité* de déplacement dans l'espace des descripteurs (symbolisée par un cercle rouge). Dans l'exemple illustré, nous constatons un entrelacement des champs de déplacements pour les trois points présents sur le motif répété ; il n'est alors potentiellement pas possible d'obtenir avec certitude le couple de points origine - destination.

Lorsque ces points problématiques se trouvent en grand nombre dans l'image, ils peuvent lourdement compliquer la tâche de l'algorithme utilisé dans l'étape suivante, soit en augmentant le temps nécessaire à la réalisation correcte de l'objectif, soit dans le pire des cas en l'entrainant dans un cas dégénéré et donc provoquant un résultat faux. Nous présentons maintenant diverses méthodes utilisées pour pallier ce problème que nous divisons en deux catégories : celles *en amont*, c'est à dire avant l'opération d'appariement à l'inverse de celles *en aval* ; pendant et après la mise en correspondance.

3.1.2 Méthodes *en amont*

Les méthodes en amont ont l'avantage de s'attaquer au problème le plus tôt possible ; elles sont ainsi tendance à être plus efficace si l'application se prête bien aux conséquences des stratégies qu'elles utilisent. Nous pouvons passer en revue des méthodes illustrant les différentes approches utilisées habituellement.

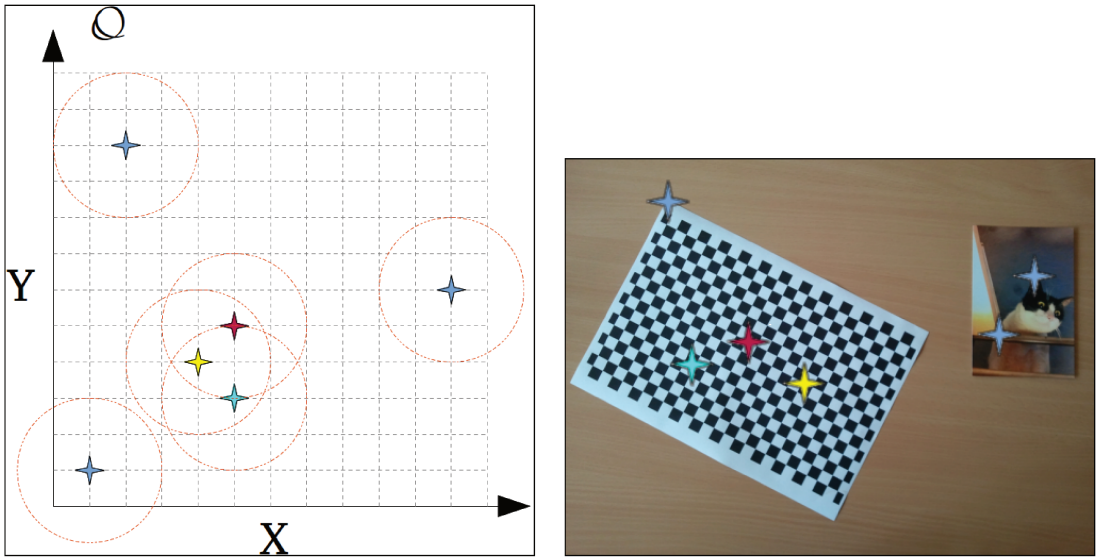


FIGURE 3.1 – A droite : illustration d'une scène présentant des motifs confusifs (le damier) ainsi qu'une zone d'intérêt (la photographie) avec des points d'intérêt dont les emplacements sont représentés par des croix.
A gauche : Représentation de ces points d'intérêt dans un espace des descripteurs imaginaire Q à deux dimensions. Les cercles en rouge correspondent aux déplacements potentiels des vecteurs lors de transformations géométriques liées à une nouvelle capture.

CAKE : Context Aware Keypoint Extraction (2012)

Bien souvent les détecteurs de points d'intérêt reposent sur une analyse a priori de l'image selon les propriétés visuelles que doivent présenter les zones saillantes. CAKE part dans une direction différente en assumant que celles-ci peuvent être vues comme des emplacements ayant une faible probabilité d'apparition et en n'établissant aucune hypothèse sur le contenu de l'image [Martins et al., 2012]. Cet algorithme utilise donc des outils issus des théories de la statistique et de l'information. En cela son approche est voisine de la nôtre que nous présenterons plus loin.

L'approche CAKE est basée sur le calcul de la quantité d'information de Shannon associée à un symbole s : $\mathrm{I}(s) = -log(p(s))$ où $p(.)$ représente la probabilité d'apparition d'un symbole. Dans notre cas, un premier raisonnement naïf serait de rattacher cette notion de symbole à un pixel x d'une image, cependant cet angle d'attaque n'est que peu pertinent étant donné la faible quantité d'information que porte ce pixel en lui même. Les auteurs proposent donc d'étudier plutôt une région w centrée sur x, $w(x) \in \mathrm{R}^{\mathrm{D}}$. Ainsi, $w(x)$ peut être considéré comme un échantillon d'une fonction densité de probabilité multi-variée. Cette approche permet d'utiliser des estimateurs par noyaux pour calculer la fonction sous-jacente, les auteurs choisissent la méthode de Parzen puisqu'il nécessite uniquement un nombre suffisant d'échantillons pour fonctionner.

Ainsi, la probabilité d'un échantillon $w(y)$ devient :

$$\hat{p}(w(y)) = \frac{1}{\mathrm{N}h} \times \mathrm{K}\left(\frac{d(w(y), w(x))}{h}\right) \tag{3.1}$$

avec d une distance, K un noyau, h une fenêtre et N le nombre de pixels présents dans l'image. L'application de cette méthode permet d'atténuer la contribution des échantillons x en les diffusant dans un voisinage de R^{D} selon les propriétés (la forme de dispersion) définies par le noyau K qui est ici un classique noyau gaussien. Le paramètre h du

terme de droite devient alors une déviation standard σ_k et celui de gauche une constante Γ pour encadrer les valeurs obtenues.

Tout cela permet aux auteurs de définir leur mesure de saillance comme étant :

$$m(y) = -log\left(\frac{1}{N\Gamma} \times e\left(-\frac{d^2(w(y), w(x))}{2\sigma_k^2}\right)\right) \tag{3.2}$$

Et les points d'intérêt correspondent alors aux maximums locaux de m qui sont au dessus du seuil Γ.

La distance d utilisée est celle de Mahalanobis. Si W est l'ensemble des $w(x)$ et Σ_W la matrice de covariance de W, la distance de Mahalanobis entre $w(y)$ et $w(x)$ est :

$$d_M(w(x), w(y)) = \sqrt{(w(x) - w(y)^T \Sigma_W^{-1} (w(x) - w(y))} \tag{3.3}$$

Enfin, la sélection de la valeur de la déviation σ_k est délicate, elle ne doit être ni trop grande ni trop petite pour garantir un lissage permettant d'estimer correctement. Les auteurs proposent d'estimer σ_k^*, le σ_k optimal à l'aide de distributions univariés :

$$\sigma_k^* = \arg\max \int w_i w_{i+1} \frac{1}{\sqrt{2\pi}\sigma} \left| \frac{d\left(e^{\frac{-(w-w_i)^2}{2\sigma^2}} + e^{\frac{-(w-w_{i+1})^2}{2\sigma^2}}\right)}{dw} \right| dw \tag{3.4}$$

où w_i et w_{i+1} sont les paires les plus éloignées d'échantillonnages consécutifs dans la distribution. La résolution de cette équation montre que $\sigma_k^* = |w_i - w_{i+1}|$.

Il y a plusieurs façons possibles de définir les régions échantillons $w(x)$. Celle proposée par les auteurs repose sur l'utilisation des matrices hessiennes. Comme nous l'avons vu précédemment, leurs propriétés permettent de bien décrire les caractéristiques de la forme locale de l'image. Nous avons ainsi, pour une image L lissée à l'aide d'un filtre gaussien :

$$\begin{aligned} w(x) = [&t_1^2 L_{xx}(x; t_1) \quad t_1^2 L_{xy}(x; t_1) \quad t_1^2 L_{yy}(x; t_1) \quad t_2^2 L_{xx}(x; t_2) \quad t_2^2 L_{xy}(x; t_2) \quad t_2^2 L_{yy}(x, t_2) \\ &\ldots t_M^2 L_{xx}(x; t_M) \quad t_M^2 L_{xy}(x; t_M) \quad t_M^2 L_{yy}(x; t_M)] \end{aligned} \tag{3.5}$$

Avec L_{xx}, L_{xy} et L_{yy} les dérivés partielles d'ordre 2 de L et les t_i les paramètres d'échelle.

Pour conclure, CAKE présente toutefois le défaut de ne pas prendre en compte un calcul d'orientation. De plus, l'ensemble de points retournés n'est pas adapté pour une demande de points d'intérêt denses (comme l'appariement stereo ou le suivi d'objet), mais pour la description d'image étant donné qu'il s'efforce d'en extraire les informations les plus pertinentes.

Descripteurs enrichis

Une autre approche consiste à ajouter dans le vecteur caractéristique du descripteur du point d'intérêt une information supplémentaire sur le voisinage local du point pouvant servir à enlever l'ambiguïté. Un exemple est la contribution de Mortensen *et al.* avec

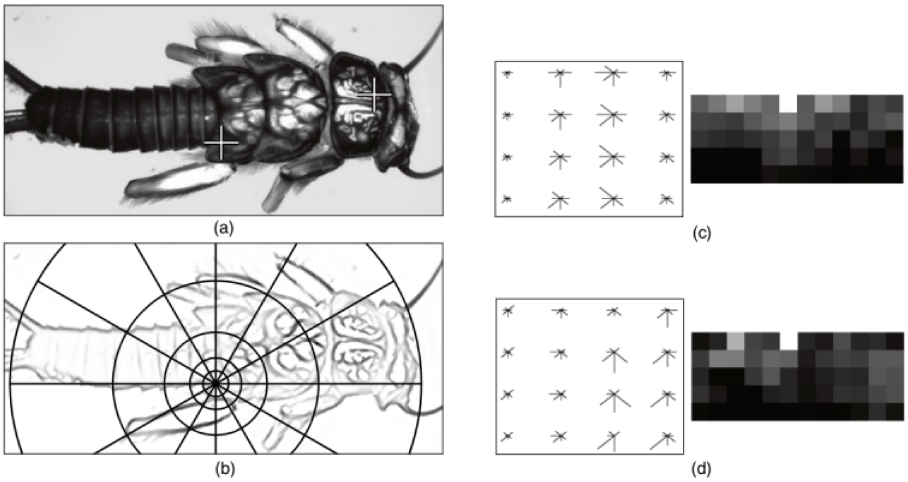


FIGURE 3.2 – Illustration de SIFT enrichi avec *shape context* pour une meilleure prise en compte du contexte local. (a) image exemple avec deux points d'intérêt (b) image des courbures avec calcul du *shape context* pour le point d'intérêt au centre (c-d) descripteurs SIFT et *shape context* des deux points d'intérêts de l'image. [Mortensen et al., 2005]

[Mortensen et al., 2005] où à l'aide d'un histogramme de *Shape Context* les auteurs augmentent le descripteur SIFT qui devient alors :

$$F := \begin{bmatrix} \omega L \\ (1-\omega)G \end{bmatrix} \tag{3.6}$$

où L correspond au vecteur SIFT classique, G un histogramme shape-context de 60 entrées et ω un paramètre de pondération.

Comme indiqué, G est aussi un histogramme, cependant son but est de représenter le nombre de pixels correspondant à des bords selon une représentation log-polaire qui couvre une partie relativement importante de l'image. Afin d'éviter l'écueil d'un simple calcul de bords qui pâtirait grandement d'un changement de contraste, ceux-ci sont obtenus en se basant une fois de plus sur les matrices hessiennes : le maximum de courbure est retourné à l'aide de la plus grande valeur absolue des valeurs propres de la matrice. Les dérivés d'ordre 2 sont calculées par convolution de l'image avec les dérivées d'une gaussienne d'échelle σ ($\sigma = 2$). Ainsi, à partir de la matrice Hessienne :

$$H(x,y) = \begin{bmatrix} r_{xx} r_{xy} \\ r_{xy} r_{yy} \end{bmatrix} = I(x,y) \times \begin{bmatrix} g^{\sigma}_{xx} g^{\sigma}_{xy} \\ g^{\sigma}_{xy} g^{\sigma}_{yy} \end{bmatrix} \tag{3.7}$$

L'information de courbure de l'image est obtenue avec :

$$C(x,y) = |\alpha(x,y)| \tag{3.8}$$

où $|\alpha(x,y)|$ est la plus grande valeur propre de H. La figure 3.2(b) montre l'image de courbure obtenue et utilisée pour la construction de l'histogramme du contexte.

Ainsi, lors du calcul d'un vecteur descripteur, les valeurs de l'image des courbures sont accumulées dans les entrées correspondantes de l'histogramme log-polaire de taille 5×12. Son diamètre est égal à la diagonale de l'image afin de prendre en compte l'ensemble de l'information contextuelle. On remarque bien évidemment que les cellules sont plus

nombreuses et à granularité plus fine autour du point d'intérêt étudié cependant chaque entrée est pondérée par une gaussienne inversée :

$$w(x,y) = 1 - e^{-\frac{((x-x_j)^2+(y-y_j)^2)}{2\sigma^2}} \tag{3.9}$$

Pour être en cohérence avec le vecteur SIFT classique utilisé, l'histogramme est normalisé afin d'être invariant au changement de contraste. Enfin, après avoir observé qu'il n'était pas nécessaire de travailler avec une définition d'image aussi grande que celle étudiée pour analyser l'information contextuelle, les auteurs proposent de réduire l'image des courbures d'un facteur 4 avec un filtre passe-bas de Haar, lissé ensuite par un filtrage gaussien ($\sigma = 3$) ; cela a en plus le mérite d'accélérer de façon non négligeable les temps de calcul.

Toutefois, le défaut de ce descripteur est qu'il nécessite de prendre en compte ses deux composantes (SIFT et shape-context) lors du calcul d'appariement. La distance de la partie SIFT est obtenue de façon tout à fait classique selon une distance linéaire d_L tandis que l'histogramme du contexte nécessite une mesure du χ^2 :

$$d_G = \chi^2 = \frac{1}{2}\sum_k \frac{(h_{i,k} - h_{j,k})^2}{(h_{i,j} + h_{j,k})^2} \tag{3.10}$$

Le but est de normaliser les entrées de l'histogramme qui sont spatialement les plus grandes afin que les petites différences entre ces entrées produisent une plus petite distance que les différences entre les petites entrées.

Cette considération nous donne le calcul de distance final suivant :

$$d = \omega d_L + (1-\omega) d_G \tag{3.11}$$

avec ω le coefficient de pondération présenté lors de la définition de F.

Filtrage par histogrammes

Avec une approche dont l'objectif est relativement similaire à celle que nous allons proposer dans la section suivante, Chazalon *et al.* proposent une méthode de filtrage de points d'intérêt afin de ne conserver qu'un noyau dur représentatif [Chazalon et al., 2015]. L'application dans laquelle ils se projettent est celle de l'identification et segmentation d'un document dans un flux vidéo, assez classique d'un point de vue vision par ordinateur : pour chaque image du flux vidéo, un ensemble de descripteurs locaux est calculé et mis en correspondance avec des ensembles de descripteurs d'images de documents que l'on souhaite retrouver.

Bien évidemment, même si ce scénario est classique dans la communauté Vision, il est tout sauf trivial lorsqu'il s'agit de travailler avec des images de documents. Les auteurs ont le même constat que le nôtre quant à l'inadéquation des descripteurs locaux de points d'intérêt. Ils proposent donc d'analyser pour chaque image modèle l'ensemble des descripteurs afin de ne conserver que les plus pertinents. C'est donc un filtrage qui nécessite une étape d'entrainement ; pour chaque image modèle, il est nécessaire d'étudier quels sont les points d'intérêts qui sont les plus utilisés lors de l'estimation d'une homographie par RANSAC [Fischler and Bolles, 1981] avec un flux vidéo d'entrainement. Les grandes lignes de leur méthodologie, illustrée par la figure 3.3 sont les suivantes :

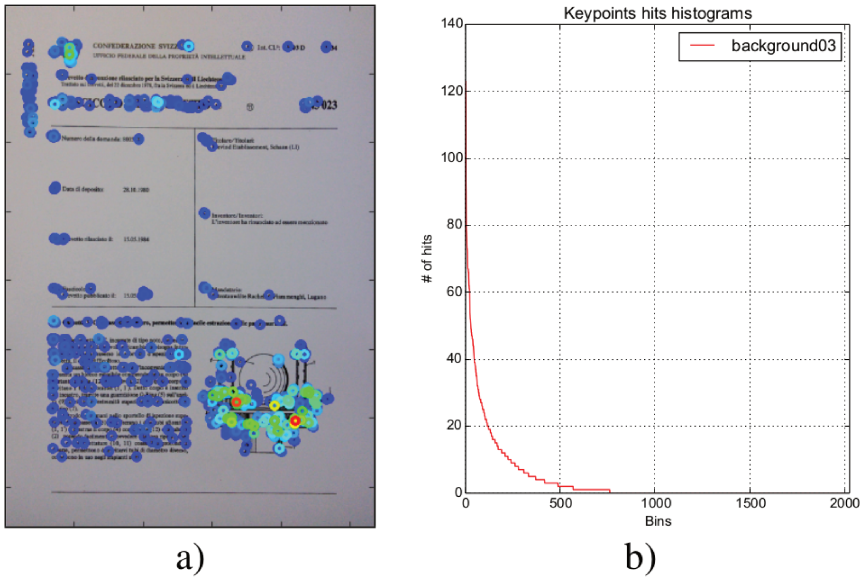


FIGURE 3.3 – Utilisation réelle des points d'intérêt d'une image modèle lors de l'estimation de l'homographie avec une occurrence dans une image issue d'un flux vidéo. a) Visualisation des points (des couleurs vives impliquent un usage plus important) b) histogramme d'utilisation de ces points. [Chazalon et al., 2015]

1. extraction d'un ensemble de points K_i d'un modèle d_i,
2. construction de l'histogramme $h_{i,j}$ de l'usage des points d'intérêt lors de la mise en correspondance de d_i avec chaque image d'entrainement
3. construire un ensemble de points $\mathrm{K}_i|t$ avec t points d'intérêt qui possèdent les plus hautes valeurs dans $h_{i,j}$

Nous constatons que cette méthode est particulièrement efficace. Si les vidéos d'entrainement sont assez diverses et nombreuses, les ensembles K_i/t sont très pertinents et permettent d'améliorer significativement la qualité de la segmentation ainsi qu'une réduction importante du temps de traitement en raison d'une convergence plus rapide de RANSAC. En revanche, bien que ne nécessitant pas de vérité-terrain, l'obligation d'avoir recours à cette étape d'entrainement reste contraignante. De plus, le filtrage ne s'opère que sur les images modèles ; nous avons donc toujours des ensembles de points potentiellement lourds et confusifs lors du traitement du flux vidéo. Comme nous le verrons lors de la présentation de notre proposition dans la section suivante, notre contribution tente d'apporter plus de souplesse d'utilisation.

Cohérence spatiale

Dans [Rusiñol and Lladós, 2009], Rusiñol *et al.* explorent l'utilisation de modèles *bag-of-words* dans une application de classification de documents par détection de logo. Les mots sont formés à partir de descripteurs locaux (les auteurs utilisent SIFT et Shape-Context). A partir de n_i descripteurs, un logo L_i est représenté par :

$$\mathrm{L}_i = \{(x_k, y_k, s_k, \mathrm{F}_k)\} \tag{3.12}$$

avec x et y les coordonnées du point d'intérêt, s son échelle et F son vecteur. Ensuite, à partir d'un document D_j, le calcul d'une mesure de correlation entre un point d'intérêt et

un logo L_i se fait à partir du ratio suivant :

$$M(L_i, D_j^q) = \frac{N_1(L_i, D_j^q)}{N_2(L_i, D_j^q)} \tag{3.13}$$

avec

$$N_1(L_i, D_j^q) = \min_k (F_q - F_k) \tag{3.14}$$

$$N_2(L_i, D_j^q) = \min_{k \neq N_1(L_i, D_j^q)} (F_q - F_k) \tag{3.15}$$

Avant d'enregistrer dans un histogramme les résultats pour sélectionner quel logo a été retrouvé, les auteurs proposent un filtre très simple pour forcer une certaine cohérence spatiale lors de l'examen d'une image : les points isolés sont signalés comme faux positifs et sont simplement écartés. C'est une forme de traitement de la confusion. Comme nous allons voir très bientôt, cette idée de cohérence spatiale est aussi présente dans une extension de RANSAC.

3.1.3 Méthodes *en aval*

Par « méthodes en aval » nous exprimons celles qui tentent de compenser le problème de la confusion au moment d'établir l'appariement entre les descripteurs ou lors de l'utilisation des correspondances établies. Il s'agit de minimiser les conséquences de la confusion lorsque l'on a conservé des ensembles de descripteurs problématiques.

Test du Ratio

Une méthode assez simple d'utilisation et recommandée dans presque tous les cas de figure est le test du ratio tel que présenté par Lowe dans [Lowe, 2004]. Il s'agit, lors de la tentative d'appariement d'un descripteur de point d'intérêt, de rechercher les deux plus proches voisins dans l'ensemble d'arrivée et non plus le candidat le plus proche comme dans une simple recherche par méthode « force brute ».

En calculant le rapport de distance entre ces deux plus proches voisins nous obtenons un indice de confiance nous renseignant sur la qualité de cet appariement. En effet, bien souvent, une forte différence entre les deux plus proches voisins doit pousser à envisager le cas d'un mauvais appariement car après tout une simple recherche force brute ne fait que trouver le descripteur le plus proche qui peut en réalité être totalement erroné dans le cas où il n'existe tout simplement pas de correspondant. Or, dans la mesure où traditionnellement les détecteurs de point d'intérêts retournent un nombre conséquent d'emplacements dans l'image avec une forte concentration dans les zones à haute fréquence dans un souci de répétabilité, les deux plus proches voisins sont en règle générale relativement proches dans l'espace des descripteurs.

Cette méthode sera désignée par l'acronyme 2NN (*two nearest neighbours*) dans la suite du document. Le seuil utilisé est traditionnellement compris entre 0.6 et 0.8.

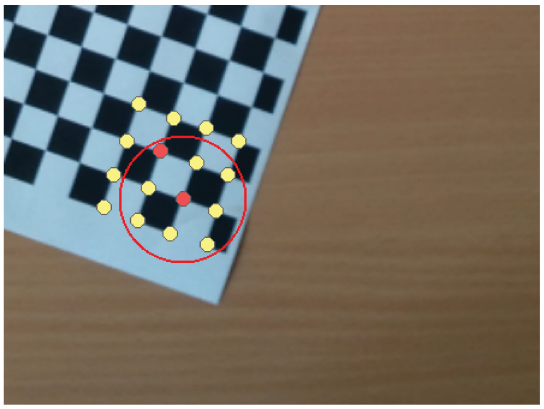


FIGURE 3.4 – Notion de voisinage spatial des points d'intérêt : l'ambiguïté peut-être levée en prenant en compte un voisinage plus étendu afin de tirer parti d'une information contextuelle.

SCRAMSAC

SCRAMSAC [Sattler et al., 2009] est une modification de l'algorithme RANSAC [Fischler and Bolles, 198 qui consiste à rajouter un test de cohérence spatiale avant le calcul du modèle de transformation. L'idée sur laquelle se base cette méthode est qu'il est possible de réduire l'ambiguïté des descripteurs en prenant en considération un voisinage local plus étendu autour de son point d'intérêt source, comme illustré par la figure 3.4.

Pour cela, les auteurs commencent par définir pour une image I_i l'ensemble des descripteurs comment étant les quadruplets $F(I_i) = \{(x_j, y_j, \sigma_j, d_j)\}$ avec x_j, y_j la position du pixel du point d'intérêt source, σ_j son échelle et d_j le vecteur descripteur associé. Nous pouvons ensuite pour chaque descripteur f_j construire une notion de l'ensemble des descripteurs voisins $N_{I_i}(f_j)$:

$$N_{I_i}(f_j) := \left\{ f_k \in F(I_i) \{f_j\} \mid ||(x_k - x_j, y_k - y_j)||_2 \leqslant r\sigma_j \wedge S_{min} < \frac{\sigma_k}{\sigma_j} < S_{max} \right\} \tag{3.16}$$

Autrement dit, les descripteurs « voisins » sont ceux présents dans une certaine proximité spatiale relative à l'échelle du point d'intérêt ($r\sigma_j$, avec $4.5 \leqslant r \leqslant 7.5$ mais en principe souvent $r = 7$) et dont les échelles sont voisines ($S_{min} = 0.5$ et $S_{max} = 2$).

Ainsi, pour un couple d'images (I_1, I_2) d'observations d'une même scène, après appariement des descripteurs ceux-ci sont seuillés selon leur qualité afin que chaque descripteur de I_1 ait au plus un correspondant dans I_2. Ce seuillage définit l'ensemble C des correspondances :

$$C := \{(f^1, f^2), \mid f^1 \in F(I_1) \wedge f^2 \in F(I_2)\} \tag{3.17}$$

Sur cet ensemble C, il est introduit à nouveau une notion de voisinage pour une correspondance c :

$$N(c) = \left\{ (f^1, f^2) \in C \mid f^1 \in N_{I_1}(f_j^1) \wedge f^2 \in N_{I_2}(f_k^2) \right\} \tag{3.18}$$

Toutes ces définitions permettent de poser le test de « cohérence spatiale » au cœur de cet algorithme, dont l'expression est la suivante :

$$|N(f_j^1)| = |\left\{ (f^1, f^2) \in C \mid f^1 \in N_{I_1}(f_j^1) \right\}| > 0 \wedge \frac{|N(c)|}{|N(f_j^1|} \geq \theta \tag{3.19}$$

Avec θ un seuil compris entre 0 et 1 (0.55 conseillé par les auteurs). L'application de ce test permet de filtrer les cas de dispersion d'appariements qui sont une conséquence fréquente lorsqu'un motif visuel se répète plusieurs fois dans l'image à des emplacements différents. Il a aussi tendance à réduire le nombre de correspondances dans les problèmes de très forte variations d'angles qui sont difficiles à traiter avec les descripteurs classiques en ne gardant que celles présentant un degré de confiance relativement élevé.

3.2 L'algorithme CORE

En passant en revue les différentes méthodes utilisées pour traiter le problème de la confusion, nous pouvons établir le constat suivant : les méthodes en amont bien que efficaces sont spécifiques ; il s'agit souvent de créer un nouveau détecteur ou d'enrichir un descripteur existant. Leur défaut est donc dans les premiers cas d'exclure l'utilisation d'autres détecteurs qui pourraient avoir des propriétés intéressantes selon l'application et de devoir prendre en compte les spécificités des seconds cas lors des calculs d'appariement.

Les méthodes en aval, elles, présentent l'avantage intéressant d'être génériques : elles peuvent se combiner avec différents descripteurs. En revanche leur approche se situe dans la « limitation des dégâts » en ce sens qu'elles tentent de palier les conséquences d'avoir conservé des ensembles de descripteurs problématiques. Tout ceci est schématisé par la figure 3.5.

La méthode que nous proposons ambitionne de présenter les avantages des deux approches simultanément : le travail en amont et la généricité tout en étant simple. En posant à plat le problème, nous pouvons faire le constat suivant :

1. nous avons affaire à des grands ensembles de données, comme lorsque nous appliquons le détecteur SIFT sur une image prise par la caméra d'un téléphone portable avec environ 30000 points retournés et dont les descripteurs sont de dimension 128,
2. il y a de l'incertitude quant à savoir quel déplacement vont subir les vecteurs descripteurs dans leur espace,
3. et enfin, nous possédons des informations incomplètes et ambigües sur les raisons de ces déplacements ; perturbations liées au bruit, au changement d'orientation, etc.

3.2.1 Vue d'ensemble

Considérons une image I, résultat de l'acquisition d'une scène par une caméra. Soit I′, une image résultant d'une nouvelle acquisition de cette même scène dans des conditions

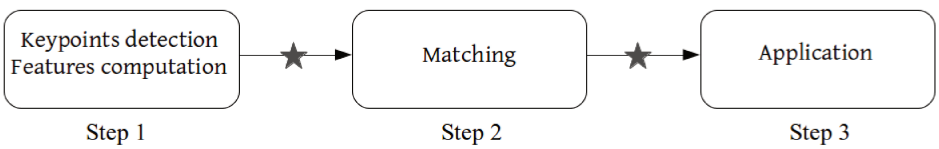


FIGURE 3.5 – File de traitement classique d'une application requérant une détection de points d'intérêt et un calcul de descripteurs en vue d'un appariement. Les symboles ⋆ correspondent aux pre/post traitements génériques.

différentes : variations d'intensité lumineuse, changement de perspective, bruit, etc. Dans notre modèle, nous considérons I comme étant déterministe tandis que I′ est une autre version, potentielle, de I que nous pouvons donc considérer comme stochastique.

Soient $\mathbf{u}_{\mathbf{i},\ \mathbf{i}\in\{\mathbf{1},\ldots,\mathbf{N}\}}$, les vecteurs descripteurs à D dimensions calculés à partir de de N points d'intérêts sur I. Leurs équivalents respectifs dans I′ sont notés $\mathbf{u'}_{\mathbf{i},\ \mathbf{i}\in\{\mathbf{1},\ldots,\mathbf{N}\}}$. Comme évoqué précédemment, une des principales caractéristiques recherchée dans un descripteur est la stabilité face aux transformations ; or il n'est pas possible d'empêcher complètement les variations des valeurs calculées, nous pouvons tout au plus les réduire. Cette approche nous permet de considérer les vecteurs $\mathbf{u}'_i$ comme des variables aléatoires et nous tentons d'établir pour chaque point d'intérêt un critère qui caractérise le risque de confusion, autrement dit, une valeur liée à la probabilité que dans l'image I′, un vecteur $\mathbf{u}'_{j,j\neq i}$ soit plus proche de $\mathbf{u}_i$ que de $\mathbf{u}_j$.

3.2.2 Calcul du critère

Pour chaque point d'intérêt i de I nous définissons une valeur C_i qui caractérise le risque de confusion, comme la densité de probabilité qu'un autre autre vecteur aléatoire $u'_{j,j\neq i}$ soit égal à $\mathbf{u}_i$, autrement dit $P_{\mathbf{u'_{j,\ j\neq i}}}(\mathbf{u_i})$.

Cette définitions nous permet d'écrire :

$$C_{\mathbf{i}} \equiv P_{\mathbf{u'_{j,\ j\neq i}}}(\mathbf{u_i}) = \sum_{\mathbf{j}\neq\mathbf{i}} \Pr(\mathbf{k}=\mathbf{j},\mathbf{u}=\mathbf{u_i}) \tag{3.20}$$

$$= \sum_{\mathbf{j}\neq\mathbf{i}} P_{\mathbf{k},\ \mathbf{k}\neq\mathbf{i}}(\mathbf{j}) P_{\mathbf{u/j}}(\mathbf{u_i}) \tag{3.21}$$

Où $P_{\mathbf{k},\ \mathbf{k}\neq\mathbf{i}}(\mathbf{j})$ correspond à la probabilité de choisir un point d'intérêt j et $P_{\mathbf{u/j}}(.)$ la fonction de la densité de probabilité d'un vecteur. Nous réalisons l'hypothèse que les probabilités de choisir un point d'intérêt sont équiprobables, ce qui nous permet d'écrire $P_{\mathbf{k},\ \mathbf{k}\neq\mathbf{i}}(\mathbf{j}) = \frac{1}{\mathbf{N}-1}$. De surcroit, nous partons du principe que $P_{\mathbf{u/j}}(\mathbf{u})$ ne dépend que d'un voisinage de $\left|\mathbf{u}-\mathbf{u_j}\right|$, et donc : $P_{\mathbf{u/j}}(\mathbf{u}) = K\left(\left|\mathbf{u}-\mathbf{u_j}\right|\right)$.

Ces hypothèses nous permettent d'estimer C_i à l'aide de l'estimateur par noyau de Parzen-Rosenblatt [Rosenblatt, 1956, Parzen, 1962] :

$$C_{\mathbf{i}} = \frac{1}{(\mathbf{N}-1)} \sum_{\mathbf{j}\neq\mathbf{i}} K\left(\left|\mathbf{u_i}-\mathbf{u_j}\right|\right) \tag{3.22}$$

Au final, cela nous permet de proposer un algorithme de réduction de la confusion que nous appelons CORE (*COnfusion REduction*, algorithme 3). Son processus est très simple, il s'agit de calculer un critère pour chaque point d'intérêt et de se servir d'une valeur seuil pour décider de le conserver ou non.

Les étapes *(a)* et *(b)* de cet algorithme sont étudiées en détail dans les sections suivantes.

Descripteurs flottants

Comme évoqué précédemment, de nombreuses causes sont à l'origine des variations dans les vecteurs de descripteurs, dont les origines peuvent être naturelles ou assimilées comme telles. Cette dernière hypothèse nous permet de définir K comme un classique noyau gaussien à D dimensions :

Algorithme 3 : l'algorithme CORE

Données : I : Image fournie
Données : p : Probabilité de confusion tolérée
Données : D : Dimension du descripteur
Données : σ : Variance moyenne d'une composante d'un vecteur descripteur (cas flottants)
μ : probabilité de basculement d'un bit dans le vecteur descripteur (cas binaire)
Données : $C_{th} \leftarrow$ calculSeuil(p, σ|μ, D) *(b)*
Résultat : χ : Point d'intérêts à conserver
K ← point d'intérêts trouvés
U ← vecteurs descripteurs associés
pour $u_i \in$ U **faire**
  $c_i \leftarrow$ estimateur(u_i, U) *(a)*
fin
pour $k_i \in$ K **faire**
  si $c_i < C_{th}$ **alors**
    ajouter k_i à χ
  fin
fin
retourner χ

$$K(\mathbf{u}) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^D \exp\left(-\frac{\mathbf{u}^2}{2\sigma^2}\right) \tag{3.23}$$

Ainsi, le calcul de notre critère C_i s'obtient tout simplement :

$$C_\mathbf{i} = \frac{1}{(\mathbf{N}-\mathbf{1})\left(\sigma\sqrt{2\pi}\right)^D} \sum_{\mathbf{j}\neq\mathbf{i}} \exp\left(-\frac{d_E(\mathbf{u_i},\mathbf{u_j})^2}{2\sigma^2}\right) \tag{3.24}$$

où $d_E(\mathbf{u_i},\mathbf{u_j}) = \sqrt{\|\mathbf{u_i}-\mathbf{u_j}\|}$ est la distance euclidienne entre deux vecteurs $\mathbf{u_i}$ et $\mathbf{u_j}$.

Descripteurs binaires

Dans le cas des descripteurs binaires, $\mathbf{u} = \left(u_{d,\ d\in\{1,\ldots,D\}}\right)$ est un vecteur binaire (composé de 0 et de 1) et nous exprimons $\mu = \Pr(u_d \neq u'_d)$ comme étant la probabilité que l'état du bit d soit différent entre deux images.

$P_{\mathbf{u}/\mathbf{j}}(\mathbf{u})$ suit alors un schéma de Bernoulli et K(.) peut s'exprimer sous la forme :

$$K(\mathbf{u}) = \prod_{d=1}^{D} \mu^{u_d}(1-\mu)^{1-u_d} \tag{3.25}$$

Ce qui nous donne le calcul du critère $C_\mathbf{i}$ suivant :

$$C_\mathbf{i} = \frac{1}{(\mathbf{N}-1)} \sum_{\mathbf{j}\neq\mathbf{i}} \prod_{d=1}^{D} \mu^{u_{\mathbf{i}d}\oplus u_{\mathbf{j}d}}(1-\mu)^{1-u_{\mathbf{i}d}\oplus u_{\mathbf{j}d}} \tag{3.26}$$

$$= \frac{1}{(\mathbf{N}-1)} \sum_{\mathbf{j}\neq\mathbf{i}} \mu^{d_H(\mathbf{u_i},\mathbf{u_j})}(1-\mu)^{D-d_H(\mathbf{u_i},\mathbf{u_j})} \tag{3.27}$$

où $u_{\mathbf{i}d} \oplus u_{\mathbf{j}d}$ représente une disjonction exclusive (XOR) entre $u_{\mathbf{i}d}$ et $u_{\mathbf{j}d}$, d_H étant une distance de Hamming.

3.2.3 Calcul du seuil

Puisque nous pouvons associer une valeur numérique liée au risque de confusion pour chaque vecteur descripteur, une méthode qui viendrait immédiatement à l'esprit pour extraitre un sous-ensemble de points d'intérêt serait de les trier selon leur valeur C_i et de garder les n premiers seulement. Cependant, nous réalisons bien vite qu'une telle solution n'est que peu pertinente : elle contraste avec le souci de généricité qui nous a guidés jusqu'ici dans le développement de cette méthode. Dans deux scènes différentes, les n premiers points n'auront pas forcément les mêmes valeurs C_i si la confusion inhérente dans l'image diffère fondamentalement. C'est pour cela que nous proposons le calcul d'un seuil C_{th} à nouveau dérivé d'un calcul de probabilités.

Descripteurs flottants

En conservant les notations précédentes, soient $\mathbf{u_i}$ et $\mathbf{u'_i}$ les vecteurs descripteurs calculés sur un même point d'intérêt **i** sur deux observations (images) différentes d'une même scène. Soient $\mathbf{v_i} = \mathbf{u'_i} - \mathbf{u_i}$, $\mathbf{v_j} = \mathbf{u'_j} - \mathbf{u_i}$, $d_\mathbf{i}^2 = \|\mathbf{v_i}\|^2$ et $d_\mathbf{j}^2 = \|\mathbf{v_j}\|^2$ où $\mathbf{u_j}$, $\mathbf{u'_j}$ sont les vecteurs caractéristiques associés à un autre point d'intérêt **j**.

Pour estimer C_{th} nous exprimons $C_\mathbf{i}$ comme une fonction de $p = \Pr(d_\mathbf{j}^2 < d_\mathbf{i}^2)$, la probabilité de confusion. Dans notre approche, p est un paramètre fixé par l'utilisateur qui permet de déterminer un taux de confusion acceptable. Afin de dériver cette relation nous avons d'abord besoin d'estimer $P_{d_\mathbf{j}^2(.)}$ (et donc $P_{\mathbf{v_j}}(.)$) qui est régi par la distribution des $\mathbf{u_j}$, $_{\mathbf{j}\neq\mathbf{i}}$. Toutefois, nous supposerons en outre que p ne dépend que du comportement de $P_{\mathbf{v_j}}(.)$ dans un voisinage relativement restreint de $\mathbf{u_i}$. Nous pouvons alors approximer $P_{\mathbf{v_j}}(.)$ par une distribution gaussienne à D dimensions $N(.;0,\Sigma_{\mathbf{v_j}})$ dont la valeur centrale $\Pr(\mathbf{v_j}=0) = P_{\mathbf{v_j}}(0) = C_\mathbf{i}$ grâce à la définition de $C_\mathbf{i}$ donnée dans la section précédente. L'élément diagonal $\sigma_{\mathbf{v_j}}$ de la matrice de covariance $\Sigma_{\mathbf{v_j}}$ est simplement relié à $C_\mathbf{i}$ en considérant la condition de normalisation de $P_{\mathbf{v_j}}(.)$:

$$C_\mathbf{i} = (2\pi\sigma_{\mathbf{v_j}}^2)^{-D/2} \tag{3.28}$$

Grâce à cette hypothèse, $P_{d_\mathbf{j}^2}(.)$ est obtenue par une distribution du χ^2 à D degrés de liberté qui peut être approchée par une loi gaussienne $N(.;E_\mathbf{j},\sigma_\mathbf{j})$ lorsque D est important. Les valeurs $E_\mathbf{j}$ et $\sigma_\mathbf{j}$ sont, classiquement, reliées aux valeurs $\sigma_{\mathbf{v_j}}$ et D par : $E_\mathbf{j} = \sigma_{\mathbf{v_j}}^2 D$ et $\sigma_\mathbf{j} = \sigma_{\mathbf{v_j}}^2\sqrt{2D}$.

Toujours en raison de nos considérations sur la nature gaussienne des valeurs de $\mathbf{u'_i}$ et en utilisant les mêmes constatations que précédemment, nous pouvons approcher $P_{d_\mathbf{i}^2}$ par une loi gaussienne $N(.;E_\mathbf{i},\sigma_\mathbf{i})$ avec $E_\mathbf{i} = \sigma^2 D$ et $\sigma_\mathbf{i} = \sigma^2\sqrt{2D}$.

Toutes ces définitions nous permettent de dérouler le résultat suivant :

$$
\begin{aligned}
p &= \Pr(d_{\mathbf{j}}^2 < d_{\mathbf{i}}^2) && (3.29)\\
&= \int_{-\infty}^{\infty}\int_{x}^{\infty} \mathrm{P}_{d_{\mathbf{j}}^2}(x)\mathrm{P}_{d_{\mathbf{i}}^2}(y)\,dy\,dx && (3.30)\\
&= \int_{-\infty}^{\infty}\int_{x}^{\infty} \mathrm{N}(x;\mathrm{E}_{\mathbf{j}},\sigma_{\mathbf{j}})\mathrm{N}(y;\mathrm{E}_{\mathbf{i}},\sigma_{\mathbf{i}})\,dy\,dx && (3.31)\\
&= \frac{1}{2} - \frac{1}{2\sigma_{\mathbf{j}}\sqrt{2\pi}}\int_{-\infty}^{\infty} \exp\left[\frac{-(x-\mathrm{E}_{\mathbf{j}})^2}{2\sigma_{\mathbf{j}}^2}\right] \times \\
&\quad \mathrm{erf}\left[\frac{x-\mathrm{E}_{\mathbf{i}}}{\sigma_{\mathbf{i}}\sqrt{2}}\right] dx && (3.32)\\
&= \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mathrm{E}_{\mathbf{i}}-\mathrm{E}_{\mathbf{j}}}{\sqrt{2(\sigma_{\mathbf{i}}^2+\sigma_{\mathbf{j}}^2)}}\right)\right] && (3.33)
\end{aligned}
$$

Le développement de ces calculs nous donne finalement :

$$\sigma_{v_{\mathbf{j}}}^2 = \sigma^2 \frac{\mathrm{D} + 2\sqrt{\gamma(\mathrm{D}-\gamma)}}{\mathrm{D}-2\gamma} \quad (3.34)$$

$$\text{avec} \qquad \gamma = 2\left(\mathrm{erf}^{-1}(2p-1)\right)^2 \quad (3.35)$$

à partir des résultats (3.34) et (3.35), le seuil C_{th} qui correspond à une valeur p spécifique nous est donné par (3.28).

Descripteurs binaires

Procédant de la même manière que pour les descripteurs flottants, nous définissons $\mathbf{v_i} = \mathbf{u'_i} \oplus \mathbf{u_i}$, $\mathbf{v_j} = \mathbf{u'_j} \oplus \mathbf{u_i}$, $d_{\mathbf{i}} = \mathtt{d_H}(\mathbf{u_i}, \mathbf{u'_i})$ et $d_{\mathbf{j}} = \mathtt{d_H}(\mathbf{u_i}, \mathbf{u'_j})$. A nouveau, nous partons du principe que p ne dépend que d'un voisinage restreint de $\mathbf{u_i}$ et nous modélisons localement $\mathrm{P}_{\mathbf{v_j}}(.)$ avec une distribution de Bernoulli :

$$\mathrm{P}_{v_{\mathbf{j}}}(\mathbf{u}) = \prod_{d=1}^{\mathrm{D}} \nu^{u_d}(1-\nu)^{1-u_d} \quad (3.36)$$

Cela nous permet d'obtenir la relation suivante qui lie $\mathrm{C}_{\mathbf{i}}$ à ν :

$$\mathrm{C}_{\mathbf{i}} = (1-\nu)^{\mathrm{D}} \quad (3.37)$$

Si nous considérons les expressions de Bernoulli de $\mathrm{P}_{\mathbf{v_i}}(.)$ et $\mathrm{P}_{\mathbf{v_j}}(.)$, $\mathrm{P}_{d_{\mathbf{i}}}(.)$ et $\mathrm{P}_{d_{\mathbf{j}}}(.)$ sont données par une distribution binomiale que nous pouvons approcher par des distributions de Poisson avec pour paramètres $\lambda_i = \mathrm{D}\mu$ et $\lambda_j = \mathrm{D}\nu$ respectivement.

La différence $d_{\mathbf{j}i} = d_{\mathbf{j}} - d_{\mathbf{i}}$ entre deux tirages aléatoires suivant une Loi de Poisson suit une distribution de Skellam [Skellam, 1946]. Ceci nous permet d'écrire :

$$\mathrm{P}_{d_{\mathbf{j}i}}(d) = e^{-(\lambda_j+\lambda_i)}\left(\frac{\lambda_j}{\lambda_i}\right)^{d/2} \mathrm{I}_d\left(2\sqrt{\lambda_j\lambda_i}\right) \quad (3.38)$$

avec I_d la fonction de Bessel modifiée. La distribution de Skellam peut être très bien approchée par une Loi normale $\mathrm{N}(.;\lambda_j-\lambda_i, \sqrt{\lambda_j+\lambda_i})$ ce qui nous amène à :

$$p = \Pr(d_{\mathbf{j}} < d_{\mathbf{i}}) \tag{3.39}$$
$$= \int_{-\infty}^{0} \mathrm{N}(x; \lambda_j - \lambda_i, \sqrt{\lambda_j + \lambda_i})\,dx \tag{3.40}$$
$$= \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mathrm{D}(\mu - \nu)}{\sqrt{2\mathrm{D}(\nu + \mu)}}\right)\right] \tag{3.41}$$

A nouveau, le déroulement de ces calculs nous permet d'obtenir :

$$\nu = \frac{2\mu\mathrm{D} + \gamma + \sqrt{\gamma(8\mu\mathrm{D} + \gamma)}}{2\mathrm{D}} \text{ si } p \in [0, 0.5[\tag{3.42}$$
$$\nu = \frac{2\mu\mathrm{D} + \gamma - \sqrt{\gamma(8\mu\mathrm{D} + \gamma)}}{2\mathrm{D}} \text{ si } p \in [0.5, 1[\tag{3.43}$$

avec γ donné par (3.35).

3.3 Experimentations

Un rapide exemple du résultat d'un filtrage par notre méthode est montré avec la figure 3.6 où nous pouvons observer des comportements intéressants : la majorité des points présents sur le damier sont jugés comme étant confusifs à l'exception de ceux présents sur les bords (bénéficiant probablement d'une information contextuelle) tandis que ceux présents sur la photographie sont conservés. Ce comportement est à nouveau confirmé sur les images issues de scènes urbaines de Zurich [Shao and Gool, 2003] où les points confusifs sont situés principalement sur les fenêtres des bâtiments. Enfin l'image de document texte montre des regroupements de points pertinents avec une concentration certaine dans des emplacements particuliers tels que des titres. Ceci tendrait à valider le comportement désiré de notre algorithme.

Afin de mieux comprendre la dynamique du seuillage des vecteurs et la distribution des valeurs C_i, nous pouvons nous référer à la figure 3.7 qui prouve la pertinence de notre approche pour le seuillage : comme expliqué précédement, des images différentes auront des réponses différentes au risque de confusion. Ainsi, dériver un seuil C_{th} à partir de p permet un filtrage cohérent.

Mais tout ceci ne sont que des observations visuelles éloignées de la rigueur scientifique. Afin de valider notre contribution nous souhaitons prouver que l'algorithme CORE réalise bien l'extraction d'un sous-ensemble de points d'intérêts qui est moins sujet au problème de la confusion. Pour cela, nous nous plaçons dans un cadre classique qui consiste à faire correspondre des paires de points d'intérêt dans un couple d'images d'observations d'une même scène.

3.3.1 Analyse de l'homographie

Nous nous basons sur une approche similaire à celle utilisée dans l'évaluation de SCRAMSAC en tentant d'estimer le modèle sous-jacent de transformation (*i.e.* homographie) d'un couple d'images. Avec celui-ci nous calculons le ratio des correspondances

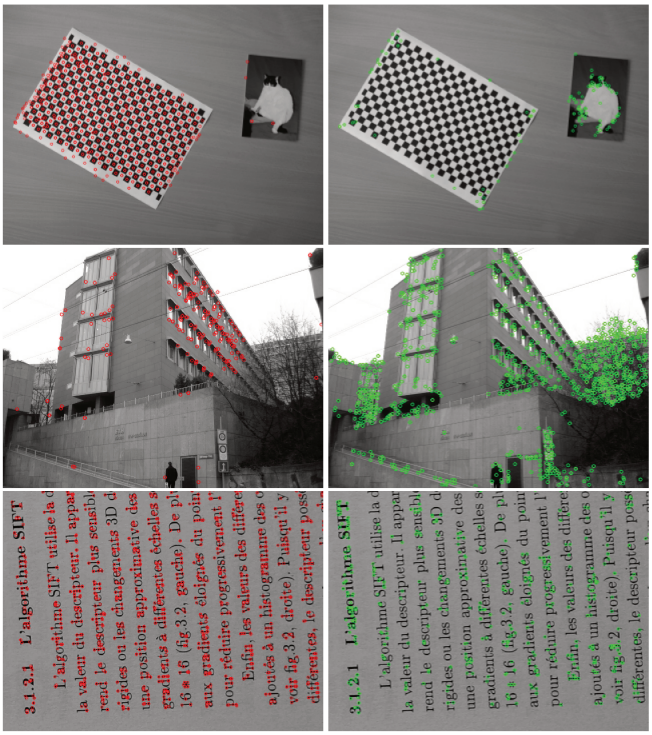


FIGURE 3.6 – Exemples de trois types d'images utilisés dans l'évaluation de notre algorithme avec application du filtrage CORE avec les points retournés par SIFT. Les ensembles enlevés sont à gauche, ceux gardés sont à droite. $p = 0.1$

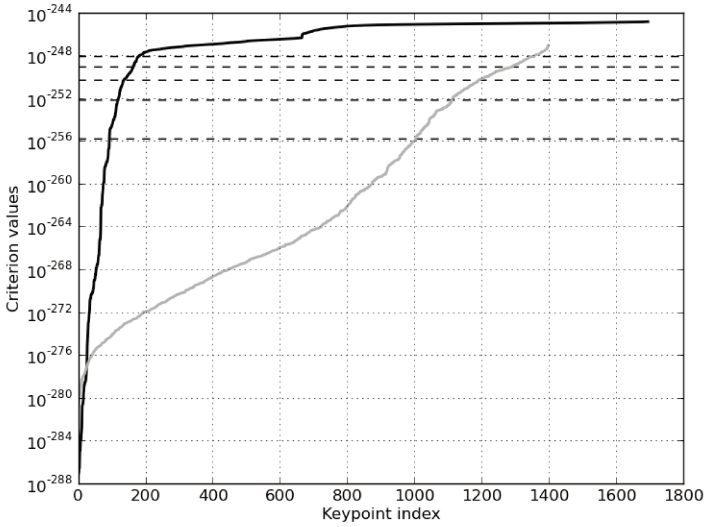


FIGURE 3.7 – Valeurs C_i triées dans l'ordre croissant des deux premières images de la figure 3.6 avec les descripteurs de SIFT, respectivement en noir et gris. Les traits horizontaux en pointillés du haut vers le bas correspondent aux valeurs seuils pour $p = 0.20, 0.15, 0.10, 0.05, 0.01$. Les points d'intérêt qui se trouvent au dessus d'un seuil fixé sont écartés.

cohérentes. Pour la plupart des expérimentations qui vont suivre nous appliquons le test du ratio de Lowe afin de conserver uniquement des correspondances de bonne qualité : nous rejetons les appariements de mauvaise qualité en calculant le ratio entre le premier et le deuxième match. Si ce ratio est en dessous d'un certain seuil (nous utilisons classiquement comme référence 0.8), le couple de descripteurs est rejeté comme étant de mauvaise qualité.

Descripteurs flottants

Nous demandons tout d'abord à un opérateur d'évaluer à la main chaque correspondance dans neuf couples d'images de la base d'images de scènes urbaines de Zurich ainsi que de deux personnels avec un motif de damier (voir figure 3.6) dans quatre scénarios différents avec l'algorithme SIFT : sans aucun filtrage (les ensembles bruts de points et de correspondances), un post-filtrage 2NN, un pré-filtrage à clustering à moyenne glissante comme utilisé par SERP [Mok et al., 2011] et un post-filtrage 2NN avec un pré-filtrage CORE ($p = 0.1$). Les résultats sont détaillés avec le tableau 3.1.

Nous observons que globalement notre contribution améliore le taux de bonnes mises en correspondances : nous avons un gain moyen d'une valeur de 8.52% pour les images de Zurich tandis que les images du damier présentent une augmentation brutale supérieure à 30%.

A partir de maintenant nous nous concentrons sur une application qui est l'estimation de la transformation sous-jacente entre deux images différentes d'une même scène à l'aide de l'algorithme RANSAC. Nous utilisons une approche similaire à celle utilisée par SCRAMSAC en évaluant la qualité de la transformation calculée avec la mesure du ratio des correspondances en conformité avec celle-ci (*inlier ratio* en anglais). Nous appliquons notre expérience suivante sur un ensemble personnel de 10 couples d'images de documents textes, issus d'articles scientifiques et capturés à l'aide de l'appareil photo d'un smartphone. Pour chaque paire d'image, nous appliquons l'algorithme CORE sur les points d'intérêt retournés par SIFT. Ceci nous donne un ensemble réduit avec lequel nous procédons à la mise en correspondance des descripteurs par force brute. Nous utilisons ensuite l'algorithme RANSAC pour estimer la matrice fondamentale de transformation et nous analysons ensuite le ratio des *inliers*. Pour une comparaison objective, nous reproduisons cette analyse avec un autre sous-ensemble de points en suivant l'idée de Lowe de la salliance basée sur une analyse du contraste afin d'avoir un ensemble de même taille que le résultat produit par CORE. Sur ces deux approches (CORE et analyse de la saillance), nous appliquons aussi le test de SCRAMSAC afin d'observer comment sa méthode de filtrage se comporte avec ces deux différentes approches. Afin d'avoir des résultats à partir d'une approche alternative, nous reproduisons ce calcul avec une méthode de filtrage basée sur un clustering à moyenne glissante (*mean-shift clustering*), utilisé par l'algorithme SERP pour détecter des motifs répétitifs dans une image.
Les résultats sont présentés avec les figures 3.9 et 3.8. Nous constatons que pour chaque valeur de p, le nombre d'inliers est toujours supérieur à ceux des autres sous-ensembles de même taille résultant d'une analyse de saillance. De surcroit, avec des valeurs p faibles (entre 0.25 et 0.05), le ratio est toujours amélioré par la méthode CORE et à partir de $p = 0.15$, même si ces processus se déroulent à des moments différents de la file de traitement, il est interessant de constater que le pré-filtrage CORE seul donne de meilleurs résultats que le post-filtrage de SCRAMSAC.

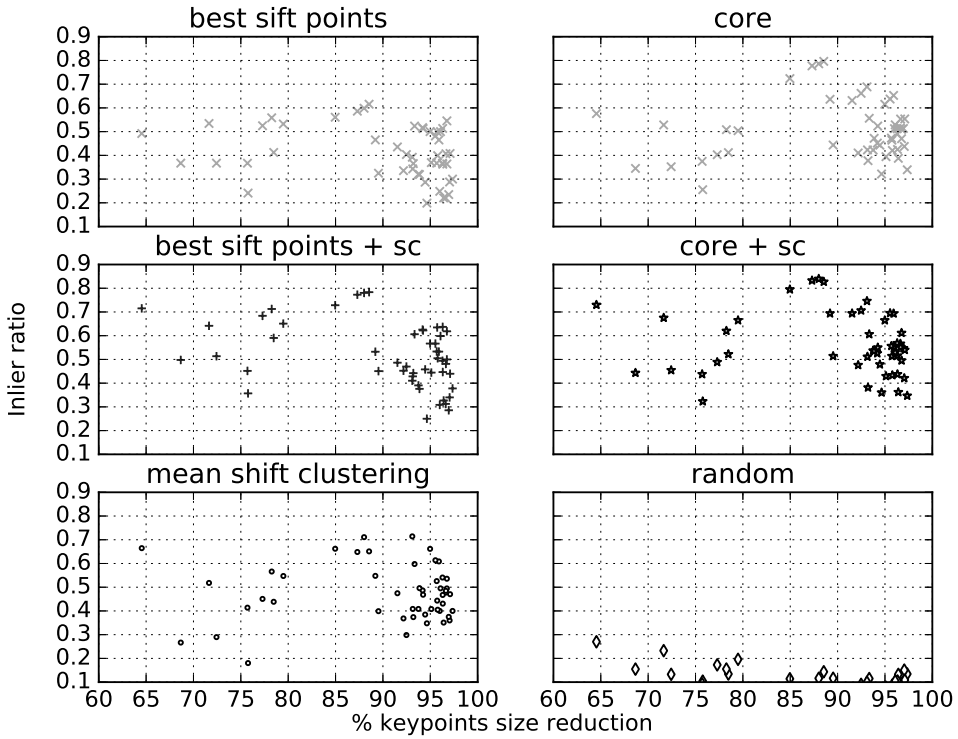


FIGURE 3.8 – Résultats individuels pour chaque couple d'images et de sous-ensemble de points d'intérêt avec les filtrages correspondances en fonction de la réduction (en %) de la taille de l'ensemble original. Chaque sous-figure correspond à une approche de filtrage et chacune est le résultat d'un couple d'image avec un sous-ensemble de points d'intérêts dont la taille est basée selon un filtrage CORE avec une valeur p spécifique.

Toutefois, pour $p = 0.50$, le ratio est en réalité plus petit avec l'algorithme CORE. Cela pourrait venir d'une trop grande confusion tolérée qui ne permet pas à la méthode d'enlever assez de points : nous ne bénéficions pas de la réduction de la confusion et des descripteurs très similaires pourraient avoir été enlevés malgré tout, alors que leur transformation n'aurait pas été suffisamment importante pour générer de la confusion. C'est pourquoi nous recommandons d'utiliser une valeur p inférieure 0.25 et les meilleurs résultats semblent être obtenus entre 0.05 et 0.10. Par ailleurs, il nous semble important de remarquer que ce pré-filtrage se comporte bien avec une phase de post-filtrage (SCRAMSAC en l'occurrence) en augmentant systématiquement le ratio, quelle que soit la valeur de p utilisée. Enfin, les résultats très médiocres du groupe de contrôle basé sur une sélection aléatoire des points nous prouvent la pertinence de notre approche par rapport à celle-ci.

Enfin, nous n'étudions pas cet aspect dans l'immédiat (voir section 3.3.2) mais il nous semble important d'évoquer un potentiel avantage de cet algorithme qui serait le gain en temps de calcul pendant la phase d'appariement des descripteurs et de l'estimation de la transformation sous-jacente entre les deux images.

Descripteurs binaires

Étant donné le fait que notre étude s'est concentrée plus particulièrement sur les descripteurs binaires et que leur popularité est croissante, nous allons approfondir l'ana-

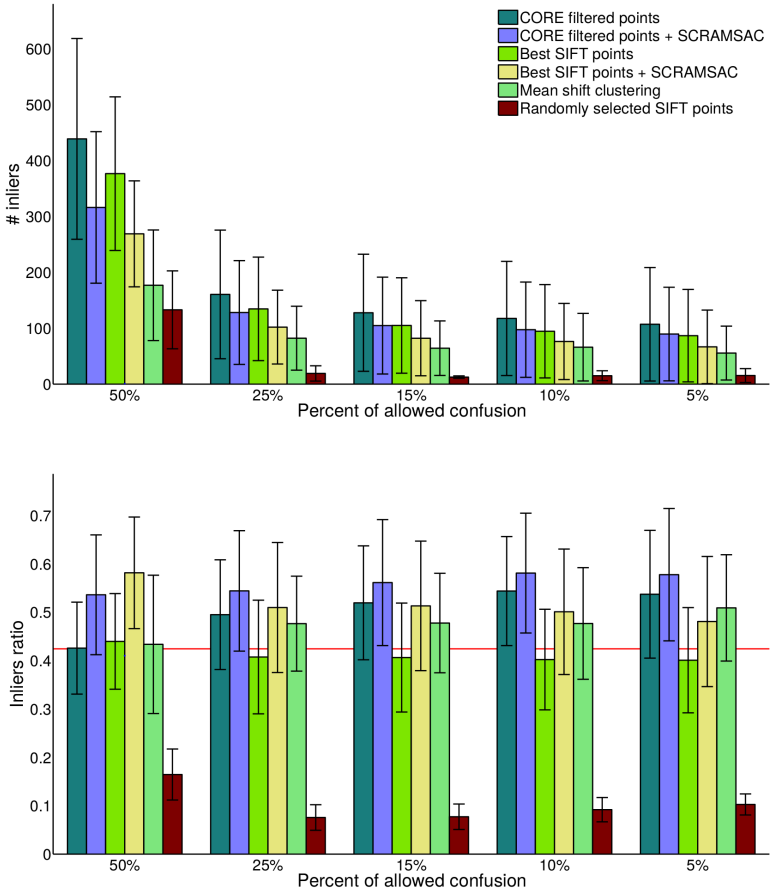


FIGURE 3.9 – Moyenne des résultats de la première partie de nos expériences. Pour chaque valeur de p, nous comparons les résultats avec des sous-ensembles de même taille. En haut : le nombre brut d'inliers, en bas : le ratio des inliers. Le trait horizontal rouge correspond au ratio de la méthode SIFT sans aucun filtrage (référence).

TABLEAU 3.1 – Comparaison des résultats (pourcentages et nombre de correspondances correctes sur le total) pour trois différentes méthodes. Dans l'ordre : simple approche SIFT, SIFT avec le test de Lowe ($d = 0.8$), regroupement par moyenne glissante avec le test de Lowe ($d = 0.8$) et CORE ($p = 0.1$) avec le test de Lowe ($d = 0.8$)

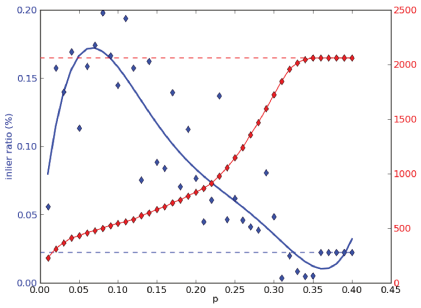
couple	unfiltered		2NN		MSC + 2NN		CORE + 2NN	
object0014	23.89%	*322 / 1348*	70.68%	*258 / 365*	67.21%	*164 / 244*	81.82%	*153 / 187*
object0008	20.00%	*336 / 1680*	52.71%	*204 / 387*	60.59%	*123 / 203*	66.51%	*143 / 215*
object0039	26.78%	*448 / 1673*	66.24%	*310 / 468*	65.74%	*192 / 289*	67.37%	*159 / 236*
object0110	24.58%	*222 / 903*	57.29%	*165 / 288*	54.24%	*83 / 153*	69.34%	*95 / 137*
object0164	25.16%	*685 / 2723*	65.66%	*545 / 830*	57.30%	*247 / 431*	71.88%	*317 / 441*
object0170	41.61%	*928 / 2230*	80.25%	*760 / 947*	79.55%	*463 / 582*	87.83%	*469 / 534*
object0181	32.35%	*645 / 1994*	74.77%	*495 / 662*	74.09%	*306 / 413*	81.69%	*290 / 355*
object0192	18.75%	*486 / 2592*	64.78%	*309 / 477*	60.16%	*225 / 374*	73.93%	*241 / 326*
object0106	25.06%	*505 / 2015*	74.71%	*325 / 435*	71.19%	*220 / 309*	77.42%	*216 / 279*
chess01	15.92%	*225 / 1413*	47.49%	*142 / 299*	75.80%	*47 / 62*	84.48%	*49 / 58*
chess02	10.72%	*182 / 1698*	35.98%	*127 / 353*	87.03%	*47 / 54*	86.44%	*51 / 59*

lyse de notre algorithme avec ces derniers. Nous choisissons quatre descripteurs classiques qui présentent une augmentation de leur complexité selon leur ordre chronologique d'apparition : BRIEF, avec son échantillonnage aléatoire des paires dans sa version par défaut, ORB avec son échantillonnage résultant d'algorithmes d'apprentissage, BRISK avec son motif réalisé à la main et FREAK, bio-inspiré. De plus, contrairement à nos tests précédents, nous allons séparer descripteurs et détecteurs, en nous basant sur trois extracteurs de points d'intérêts : SURF et BRISK sont sélectionnés car leurs méthodes d'analyses retourne un nombre conséquent de réponses, ainsi que celui de ORB car il se base sur une mesure du coin de HARRIS et son impact est par conséquent intéressant à analyser sur des documents typographiés.

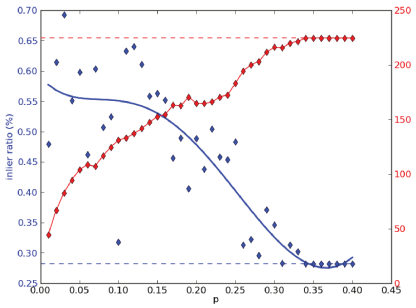
Tout d'abord, considérons une autre façon de sélectionner le paramètre *fenêtre*, μ. Pour une valeur de p fixée, nous pouvons tracer la courbe du ratio des correspondances « justes » et du nombre d'appariements conservés après filtrage de CORE comme étant fonction de μ. De hautes valeurs devraient nous indiquer de bons paramètres et puisque notre évaluation précédente nous a confirmé que les images de mire étaient d'excellents cas d'école pour l'étude de la confusion, nous porterons cette première analyse uniquement sur celles-ci avec une valeur de p très restrictive, soit 0.05 (5% de confusion tolérée). Très performant et répondant parfaitement à nos besoins, nous utilisons à nouveau l'algorithme RANSAC pour dégager le modèle de transformation sous-jacent, mais ici de trois manières différentes qui sont : simple (appariement par *brute-force*), le test du ratio de David Lowe et un cross-validation. Les résultats sont donnés dans la table 3.2. Comme nous pouvons le constater, en augmentant μ nous augmentons le nombre de points d'intérêt supprimés, améliorant ainsi le ratio ; les points d'intérêt « dangereux » sont écartés jusqu'à atteindre un extremum. A partir de ce dernier, enlever plus de points semble inefficace. Ce comportement s'explique simplement si on rappelle que μ est la probabilité de basculement d'un *bit* dans un vecteur descripteur. Ainsi, plus μ est élevé, plus nous risquons de considérer un point d'intérêt comme étant dangereux et par conséquent retiré lors du filtrage.

Cette première expérience nous renseigne sur une plage de paramètres intéressants pour μ. Grâce à cela, en fixant ce dernier, nous pouvons tracer la mesure du ratio en fonction de p. Un premier exemple est montré avec la figure 3.10 pour les images de mire avec le détecteur SURF et le descripteur ORB. Encore une fois, nous pouvons observer le comportement qui était attendu : en enlevant les points susceptibles de provoquer de la confusion, nous augmentons le ratio.

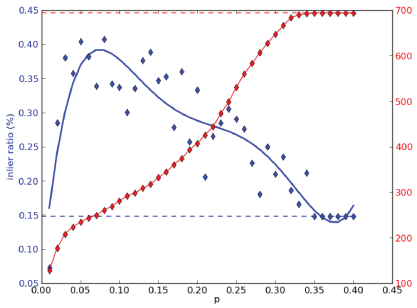
Maintenant, penchons nous plus en détails sur les quatre descripteurs et les trois détecteurs sur les images de documents en répétant la même expérience avec le filtrage 2NN avec le tableau 3.3. Une constatation particulièrement intéressante est le fait que les descripteurs ne sont pas tous égaux par rapport à la réduction de confusion en fonction du détecteur utilisé. Par exemple, BRISK partage le même comportement avec les trois détecteurs : les premiers sous-ensembles de points extraits sont toujours meilleurs que les approches sans filtrage mais l'augmentation de la valeur p provoque inévitablement une convergence du ratio vers l'approche sans filtrage. BRIEF, de son côté, donne des résultats médiocres. Une explication peut venir du fait qu'il s'agit du premier descripteur binaire moderne ; maintenant dépassé, son mécanisme simple d'échantillonnage des paires de pixels pourrait s'avérer moins discriminant dans le cas particulier des images de documents où peu d'emplacements ne ressortent vraiment par rapport aux autres. Enfin, nous



RANSAC



RANSAC avec le test de Lowe

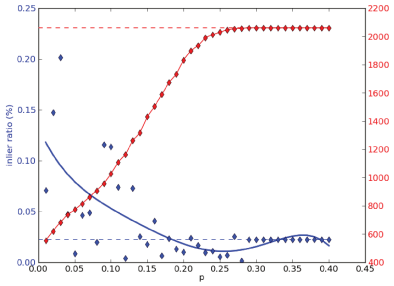


RANSAC en cross-validation

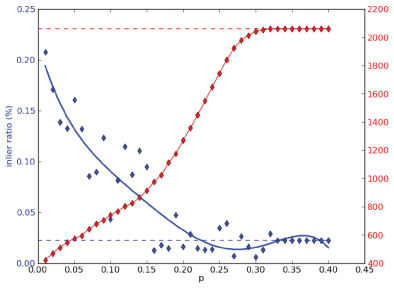
FIGURE 3.10 – Evolution du ratio des inliers (en bleu) lors de l'augmentation de p avec $\mu = 0.30$ avec le détecteur SURF et le descriptor ORB sur les images présentant un damier. Le nombre de correspondances est montré en rouge, les lignes en pointillé sont respectivement les valeurs pour les approches sans filtrage.

constatons des résultats médiocres avec le détecteur ORB qui sont pratiquement toujours en dessous de la valeur référence. Cela peut néanmoins s'expliquer : puisque ce détecteur trie les points d'intérêt en fonction d'une mesure du coin de Harris, il semble logique qu'il ne soit pas adapté à nos types d'images qui présentent de forts contrastes et de nombreux angles droits, perdant ainsi grandement son pouvoir discriminant. En réalité, parmi les descripteurs utilisés ici, seul BRISK parvient à constamment bénéficier de la réduction de confusion. Ceci pourrait être une preuve de sa grande capacité discriminante comme nous l'avons souligné dans notre synthèse du chapitre 2.

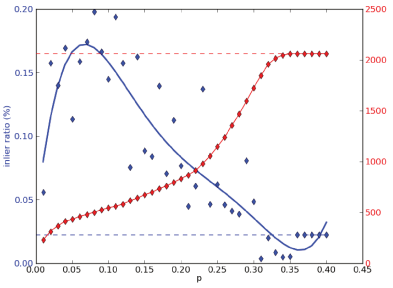
Enfin, afin de mieux comprendre l'impact du choix du paramètre μ, la figure 3.11 nous montre la conséquence de son augmentation : nous pouvons noter que la courbe du ratio semble subir un décalage pendant que le nombre de points conservés augmente progressivement. Notre évaluation semble indiquer que les valeurs intéressantes à utiliser sont comprises entre 0.20 et 0.35 mais bien entendu le choix final pourra dépendre du contexte.



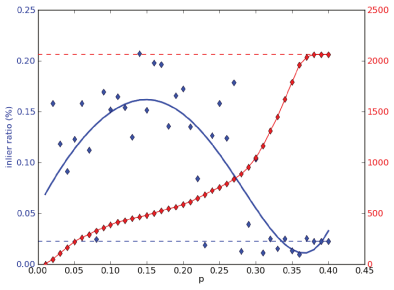
$\mu = 0.20$



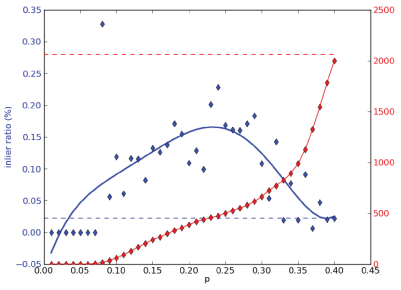
$\mu = 0.25$



$\mu = 0.30$



$\mu = 0.35$



$\mu = 0.40$

FIGURE 3.11 – Evolution du ratio des inliers en augmentant la probabilité de basculement d'un bit avec le détecteur SURF et le descriptor ORB sur les images présentant un damier.

TABLEAU 3.2 – Ratio des inliers (en bleu) en fonction de μ avec le détecteur SURF et les quatres descripteurs utilisés. Le nombre de correspondances est montré en rouge, les lignes en pointillé sont les valeurs respectives pour les approches sans-filtrages. $p = 0.05$

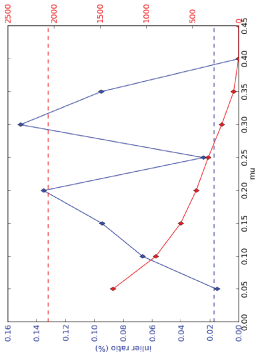
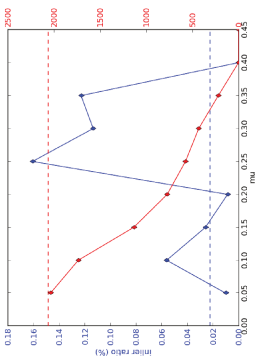
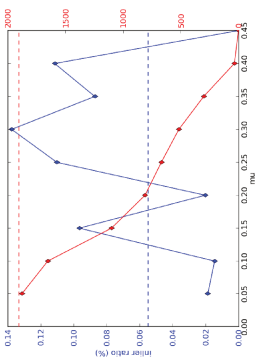
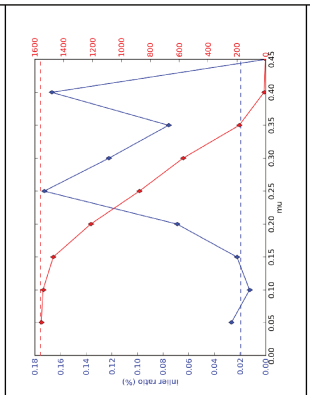
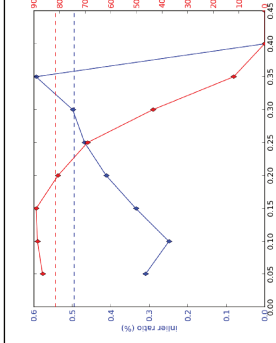
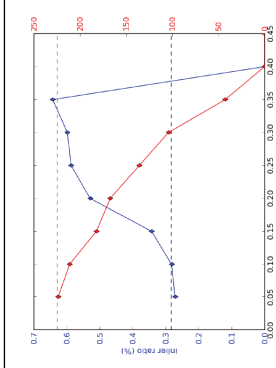
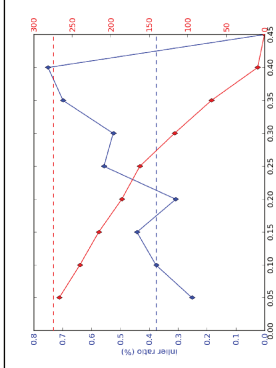
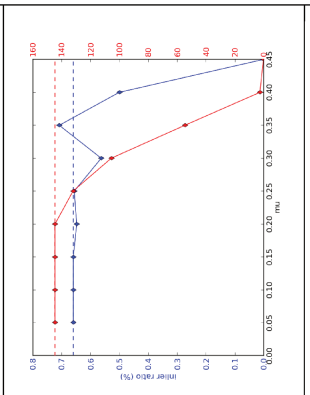
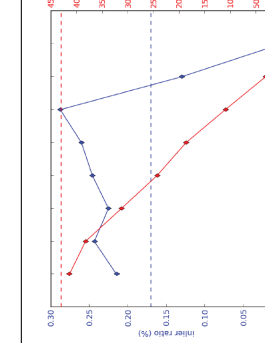
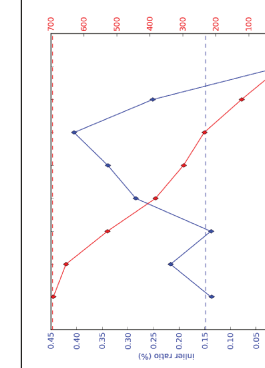
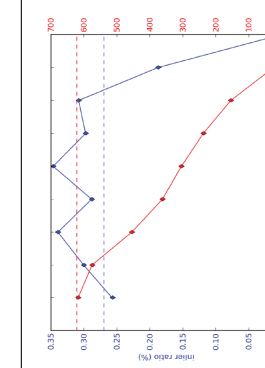
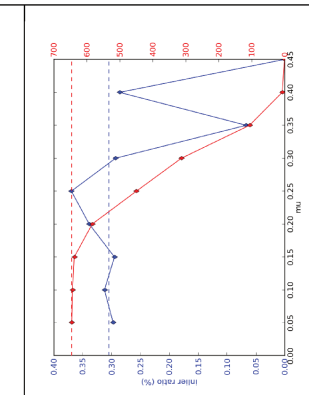
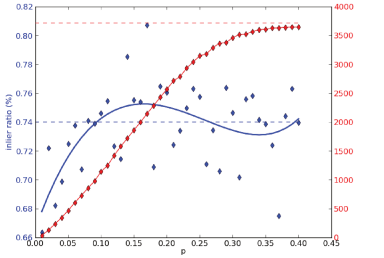
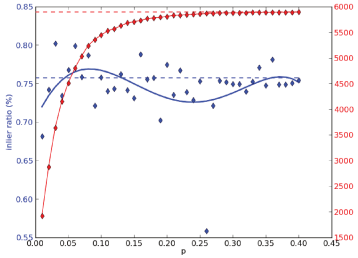
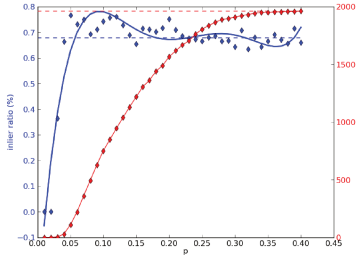
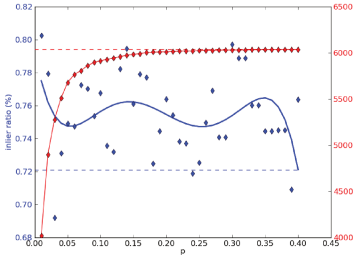
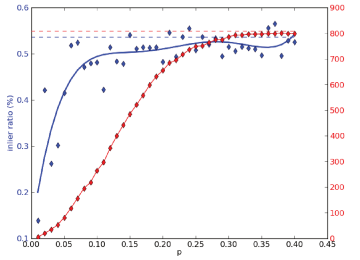
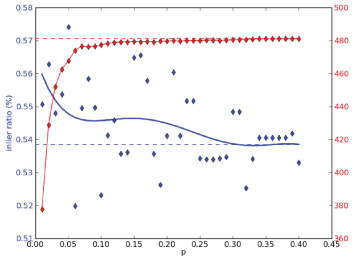
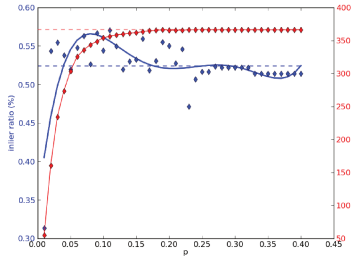
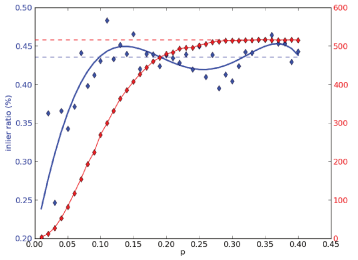
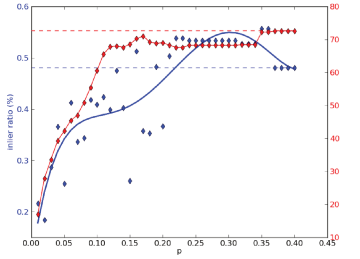
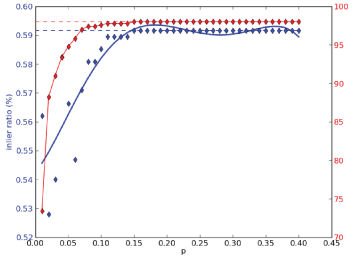
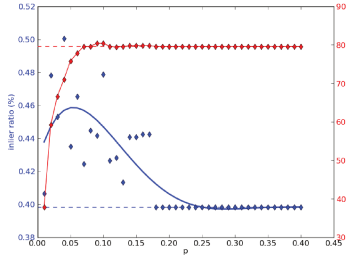
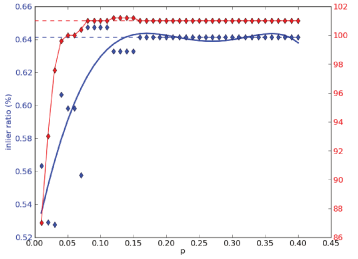
	BRIEF	ORB	BRISK	FREAK
plain				
2nn				
cross				

TABLEAU 3.3 – Ratio des inliers (en bleu) en fonction de p. Le nombre de correspondances est montré en rouge, les lignes en pointillé sont les valeurs respectives pour les approches sans-filtrages.

	BRIEF	ORB	BRISK	FREAK
SURF				
BRISK				
ORB				

3.3.2 Application à la mise en correspondance d'images de documents

Nous approfondissons maintenant l'étude du scénario de mise en correspondance d'images de documents à l'aide de descripteurs locaux dans une application plus concrète. Encore une fois, ce cas d'utilisation suit une file de traitement simple qui est le calcul de descripteurs caracteristiques locaux d'emplacements clefs dans l'image et la mise en correspondance avec un modèle.
Nous nous restreignons à un cas d'utilisation de mise en correspondance d'une unique image modèle de document à un flux vidéo. La localisation de la position précise d'une instance de l'image modèle dans chaque image de la vidéo est rendue possible en réalisant d'abord l'appariement de chaque descripteur local extrait de l'image issue de la vidéo avec les descripteurs obtenus précédement à partir de l'image modèle et indexés dans une structure de données dédiée à la recherche rapide de plus proche voisins. Encore une fois, afin d'éviter des correspondances ambigües dûes à des ensembles disjoints, nous utilisons le classique test de Lowe. L'estimation de la transformation sous-jacente entre l'image modèle et son emplacement dans l'image se fait à l'aide de l'algorithme RANSAC. Ce processus d'identification écarte progressivement de plus en plus d'informations de chaque image d'origine pour ne finalement selectionner seulement qu'un sous-ensemble consistant d'inliers qui supportent l'homographie estimée.

Filtrer les parties pertinentes des images est un procédé couteux en temps de calcul et il est donc préférable d'ajuster cette méthode pour écarter les mauvais candidats le plus tôt possible. Ce filtrage précoce est particulièrement intéressant à réaliser durant la phase d'indexation des images modèles puisqu'il n'est fait qu'une seule fois au contraire d'un filtrage des images issues du flux vidéo. Nous retenons trois approches différentes de filtrage pour établir des comparaisons :
La première est le filtrage réalisé le plus tôt possible, au niveau des propriétés visuelles immédiates du voisinage du point d'intérêt. L'objectif est de selectionner les points qui montrent la meilleure invariance en se basant sur des heuristiques basiques comme l'analyse du contraste.
La deuxième est notre algorithme CORE : l'analyse de la distribution des vecteurs caractéristiques liés aux points d'intérêt nous permet d'écarter ceux présentant un fort risque de confusion.
La troisième est appliquée plus tardivement, lors de la mise en correspondance des descripteurs et de l'évaluation de la transformation de la perspective. Elle nécessite une étape d'entrainement pour chaque modèle avec un flux vidéo dédié. Il s'agit d'écarter les descripteurs qui ne sont que peu utilisés pour l'estimation de l'homographie. De plus, la contribution de chaque descripteur peut-être pondérée par la qualité de la segmentation trouvée.

Cadre de l'évaluation

Nous évaluons les trois méthodes suivantes :
1) Référence : cette approche est un filtrage basé sur la réponse visuelle immédiate des points d'intérêt. Chaque algorithme de détection a ses propres heuristiques. Par exemple, comme dit précédemment, ORB trie les points FAST avec une mesure de Harris et SIFT se base sur une analyse du contraste. Tout cela nous permet de calculer un ensemble réduit de points d'intérêt avec une taille fixe. Nous pouvons ensuite évaluer la qualité de la mise en correspondance en réduisant progressivement les tailles des sous-ensembles de points, de 100% à 10% par décréments de 10% pour chaque étape.

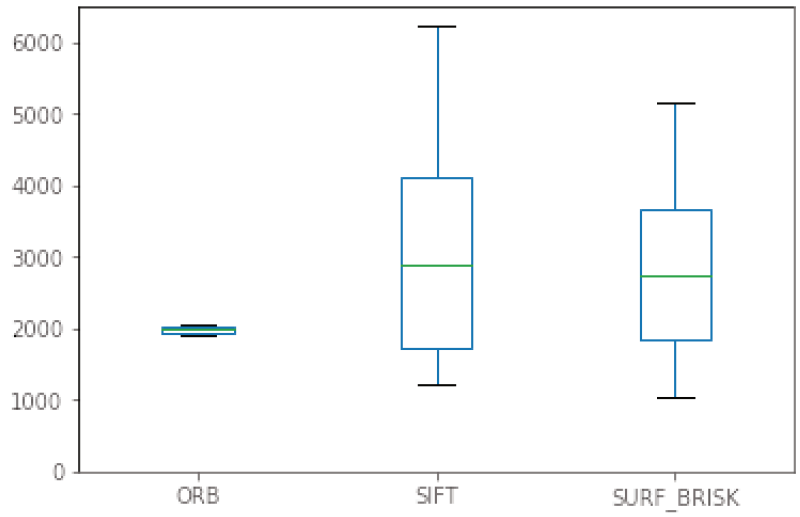
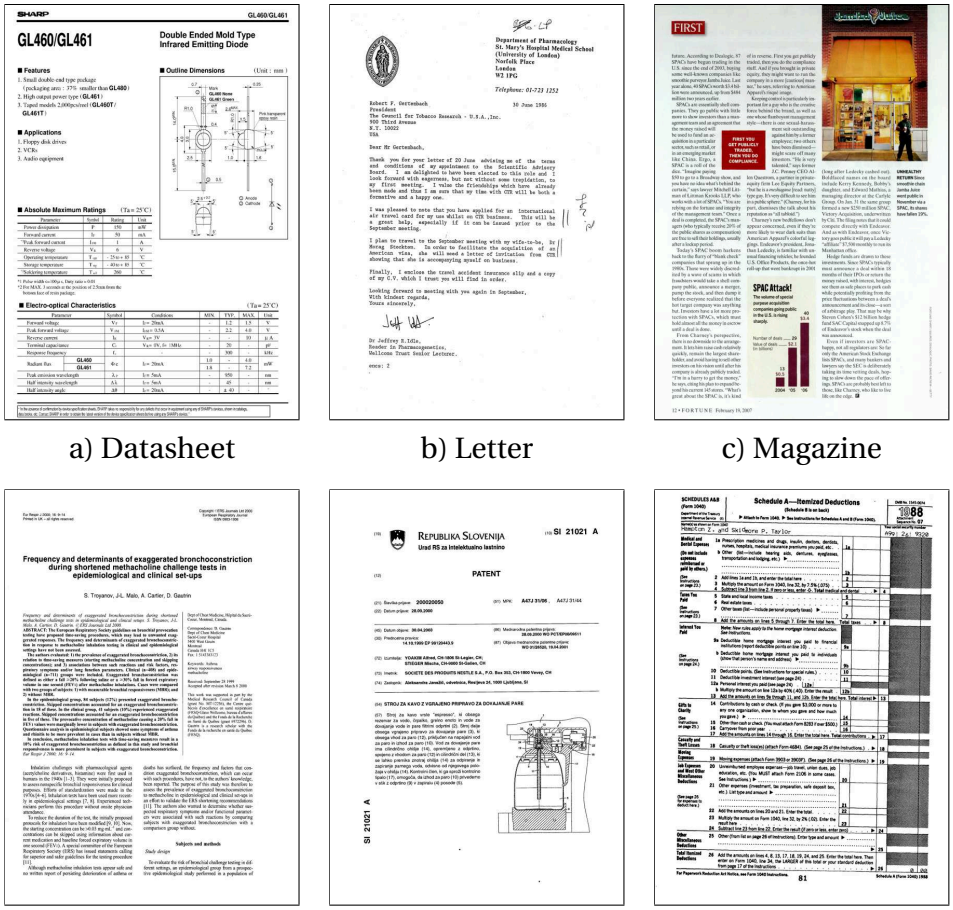


FIGURE 3.12 – Distribution des points originellement extraits de chaque image modèle pour chaque algorithme de détection.

2) Histogrames : l'optimisation par analyse d'histogrammes telle que présentée précédement. Ce filtrage repose sur une étape d'entrainement prenant en compte le nombre de fois qu'un point d'intérêt a été utilisé avec succès par RANSAC pour estimer une homographie entre l'image modèle et chaque image d'un flux vidéo d'entrainement. Tout comme la méthode de référence, cette approche nous permet de selectionner une proportion de points d'intérêt pour construire des sous-ensembles, de 100% à 10% par décréments de 10%.
3) CORE : nous ne représentons pas notre algorithme, simplement nous nous contenterons de préciser qu'au contraire de l'approche par histogrammes, l'algorithme CORE ne nécessite aucune phase d'entrainement. En revanche, sa philosophie de filtrage ne permet pas de selectionner un nombre fixe de points d'intérêts (même si nous pourrions trier les valeurs C_i et prendre les n premiers points, nous préférons rester fidèle à l'idée originale). Nous faisons varier le paramètre p de 0.15 à 0.005 pour obtenir des ensembles de points de tailles différentes.

La figure 3.12 nous montre la distribution des points originellement extraits des images modèles pour chaque technique de détection suivantes que nous utilisons :
1) ORB : ayant recours à un tri par mesure de Harris, nous paramétrons l'algorithme ORB afin d'avoir des ensembles de points d'intérêt initiaux pour chaque image d'environ 2000 points. Les résultats sont assez fidèles à ce qui est demandé puisque nous ne notons en pratique qu'une variation d'en dessous de l'ordre de 10%. 2) SIFT : les paramètres utilisés sont ceux par défaut de l'implantation d'OpenCV. Nous n'appliquons pas de restriction quant au nombre de points à retourner et nous constatons des ensembles variant entre 1000 et 6000 réponses selon l'image modèle. 3) SURF-BRISK : Suivant l'évaluation de [Rusiñol et al., 2015], nous choisissons de coupler le descripteur BRISK au détecteur SURF ; le nombre de points retournés par ce dernier est particulièrement conséquent. De la même façon, nous utilisons l'implantation d'OpenCV avec les paramètres par défaut.

Bien plus important, afin de valider le passage à l'échelle, contrairement à notre analyse précédente, nous utilisons un jeu de données robuste qui a été utilisé lors d'un concours international organisé durant ICDAR 2015, il s'agit de SmartDOC [Burie et al., 2015]. Cette



a) Datasheet b) Letter c) Magazine

d) Paper e) Patent f) Tax

FIGURE 3.13 – Extraits de documents utilisés dans SmartDOC. a) Feuille de données, b) Lettre, c) magazine (PRIMA), d) article scientifique, e) brevet, f) formulaire facture

base d'images est constituée de six types de documents différents, avec cinq documents par classe (un exemple de chacun de ces six types est illustré par la figure 3.13). Chaque document y a été capturé en vidéo avec des fonds différents à l'aide d'une caméra de tablette tactile (nexus 7), totalisant ainsi environ 25 000 images. La seule modification que nous apportons est une réduction de la taille des images modèles, afin qu'elles correspondent à la taille des occurrences présentes dans les flux vidéos.

Evaluation de la performance

Pour évaluer la performance des approches étudiées ici, nous mesurons la précision moyenne de la segmentation et la vitesse de calcul. Ces données sont normalisées afin de tenir compte de la disparité du nombre de points par image et de la durée de chaque vidéo. Nous calculons la qualité et la vitesse de calcul comme étant fonction du facteur de réduction de l'ensemble de points d'intérêts de l'image modèle pour l'approche de référence. Le facteur de réduction est lui aussi normalisé pour tenir compte de la variabilité du nombre de points filtrés par CORE, difficile à prévoir par avance.

La métrique que nous utilisons pour l'évaluation de la qualité de la segmentation est l'indice de Jaccard [Jaccard, 1901] comme utilisé dans [Chazalon et al., 2015]. Il s'agit du rapport entre l'intersection des ensembles sur l'union des ensembles. Dans notre cas, si nous considérons les quadrilatères S et G qui sont respectivement celui identifié dans

l'image et la vérité terrain pour une image f, nous avons :

$$\mathrm{JI}(f) = \frac{\mathrm{area}(\mathrm{G} \cap \mathrm{S})}{\mathrm{area}(\mathrm{G} \cup \mathrm{S})} \tag{3.44}$$

où G∩S et G∪S sont respectivement l'intersection et l'union des polygones G et S. Les valeurs sont comprises entre 0 (pire des cas) et 1 (correspondance parfaite). En pratique, les résultats en dessous de 0.6 sont la preuve d'une segmentation médiocre sans possibilité d'exploitation pour une application et il est bien sûr évident qu'il est important de rester au-dessus des valeurs de la méthode de référence.

En ce qui concerne l'évaluation du temps de calcul, nous mesurons la totalité de l'execution du processus pour chaque image de vidéo. Les méthodes de réduction pertinentes devraient à priori diminuer le délai d'execution puisque cela implique une convergence plus rapide de l'algorithme RANSAC, ayant moins de mauvais couples de descripteurs à prendre en compte qui pourraient "polluer" son analyse.

Résultats

Ceux-ci sont présentés avec la figure 3.14. Nous constatons que notre hypothèse était correcte : le temps de calcul diminue lorsque la taille des points d'intérêt des images modèles diminue et légitimise ainsi les approches employées ici. Pour chacune d'entre elles, au-delà d'un certain niveau de reduction, la qualité de la mise en correspondance diminue de façon significative. Toutefois, le seuil correspondant est différent selon la méthode employée. La méthode à histogramme reste particulièrement robuste, ce qui est censé puisque la phase d'entrainement pour selectionner les points d'intérêts nous garantie que le "noyau dur" est le plus pertinent pour la mise en correspondance. Ce n'est pas le cas du filtrage par CORE où les résultats sont en dessous des valeurs de référence lorsque la réduction devient très forte (en dessous de 15% de la taille de l'ensemble d'origine) ; ceci est certainement dû à l'aléatoire de la méthode qui ne nous garantit pas ce noyau dur avec une certitude totale. Mis à part cela, les résultats pour SIFT et BRISK sont tout à fait satisfaisants, étant au même niveau que l'approche par histogrammes lors de réductions moyennes des ensembles de points et même si la qualité chute rapidement, elle reste majoritairement supérieure à l'approche de référence, ce qui en fait une solution tout à fait pertinente, car en plus elle n'est pas supervisée.

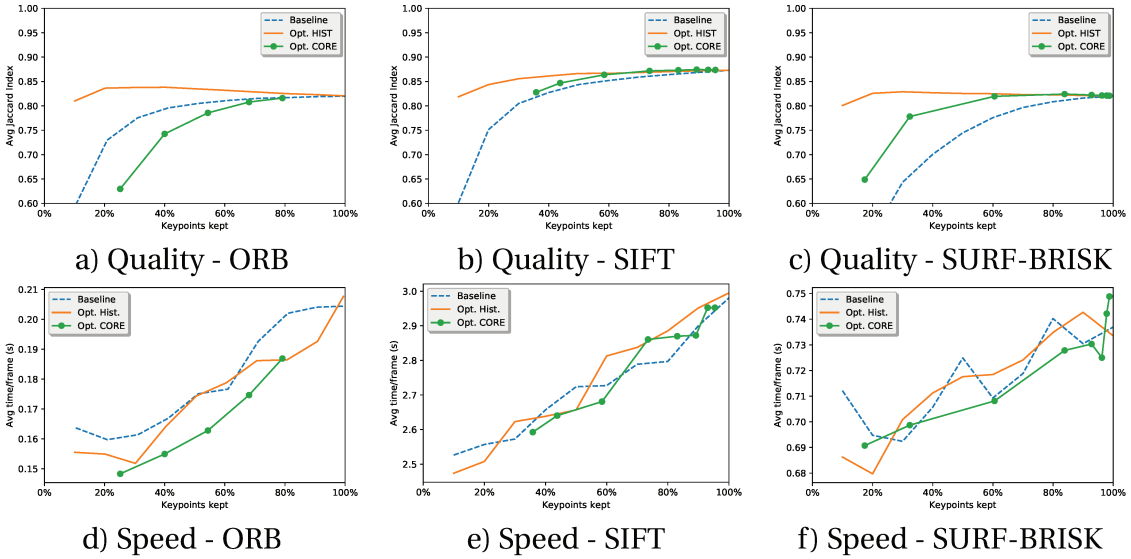


FIGURE 3.14 – Résultats de la qualité (en haut) et du temps de calcul (en bas) pour chaque méthode de filtrage de points d'intérêt (référence, histogrames et CORE) , pour trois descripteurs classiques : ORB (a, d), SIFT (b, e) et BRISK(c, f).

3.4 Optimisations

En l'état actuel, l'algorithme CORE présente un défaut non négligeable, surtout dans les scénarios de capture nomade et de traitements embarqués. Il s'agit de son coût algorithmique. Puisque le calcul du critère pour chaque vecteur nécessite de calculer sa distance par rapport à tous les autres vecteurs, la complexité algorithmique est en $O(n^2)$ pour n points d'intérêt. Ainsi, bien que peu contraignant pour des ensembles de points de tailles modérées cela devient rapidement problématique lorsque le nombre de vecteurs augmente, comme illustré par la figure 3.15. Nous explorons donc ici quelques pistes d'optimisations du temps de calcul.

Pour alléger ce coût, nous pouvons déjà nous appuyer sur la symétrie de la notion de distance : puisque la distance $D(i, j)$ est la même que la distance $D(j, i)$, si nous gardons en mémoire le calcul de l'un, nous n'avons pas à refaire le calcul de l'autre ; il s'agit de la traditionnelle dualité espace-temps en algorithmie. Il est aussi tout à fait possible de tirer parti des architectures de calcul GPU, une telle implantation est triviale puisque chaque calcul de critère est indépendant des autres. La figure 3.15 nous donne un aperçu des temps de calcul. Nous constatons qu'il est rapidement souhaitable de passer à une version parallélisée de l'algorithme.

3.4.1 GPU

Le calcul de chaque critère étant indépendant des autres, ce problème est parfaitement adapté à l'utilisation d'une architecture massivement parallèle comme les processeurs graphiques. Les *Graphics Processing Units* (GPU), ou processeurs graphiques sont des unités de traitements apparues dans les années 1980. Elles ont pour vocation de soulager le processeur principal en réalisant les calculs ayant rapport à l'affichage, principalement dans le domaine de la synthèse d'image dont l'industrie florissante du jeu-vidéo est particulièrement demandeuse.

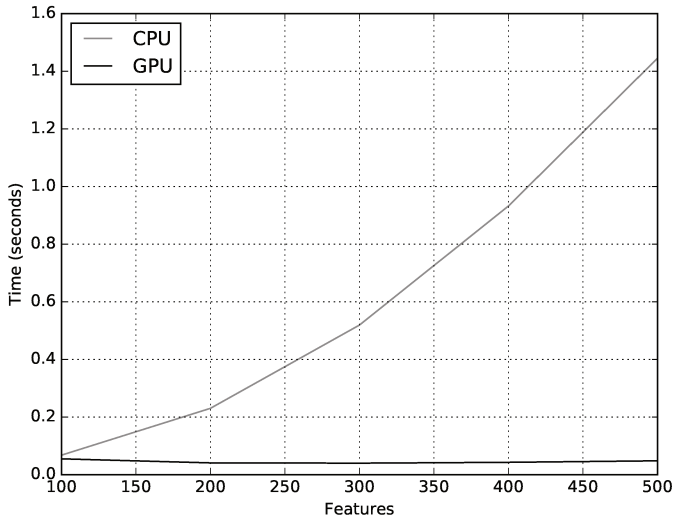


FIGURE 3.15 – Temps de calcul de la méthode proposée en fonction du nombre de points d'intérêt. Les courbes grise et noire correspondent respectivement aux implantations CPU et GPU. Le matériel utilisé est un processeur i7-6700 cadencé à 3.40 Ghz et une carte graphique Nvidia GT 640.

FIGURE 3.16 – Architectures classiques d'un CPU et d'un GPU. Le GPU se distingue par un grand nombre d'unités arithmétiques et logiques. (Nvidia)

Les processeurs graphiques

Lors d'une synthèse d'image dans un jeu-vidéo, les calculs à réaliser sont très simples mais ils sont nombreux. Pour prendre l'exemple d'un jeu typique des années 90 comme *Quake*, chaque modèle de personnage est composé d'environ 200 polygones. Lors de l'animation de ce dernier, des calculs de transformation dans l'espace en 3 dimensions sont à appliquer sur chaque point, la couleur de chaque pixel dans la scène doit être déterminée... etc. De surcroit, gardons en tête qu'il préférable d'avoir au moins 30 images rendues par seconde. Ceci explique les choix d'architecture qui ont été opérés sur ces processeurs dédiés, comme schématisé par la figure 3.16 : là où un processeur classique se caractérise par une poignée d'unités arithmétiques et logiques (ALU), le GPU doit en comporter un grand nombre pour effectuer le plus de tâches possibles en parallèle. Les ALU sont de surcroit organisés en *grappes*, chacune d'entre elle possédant sa propre structure de contrôle et son cache.

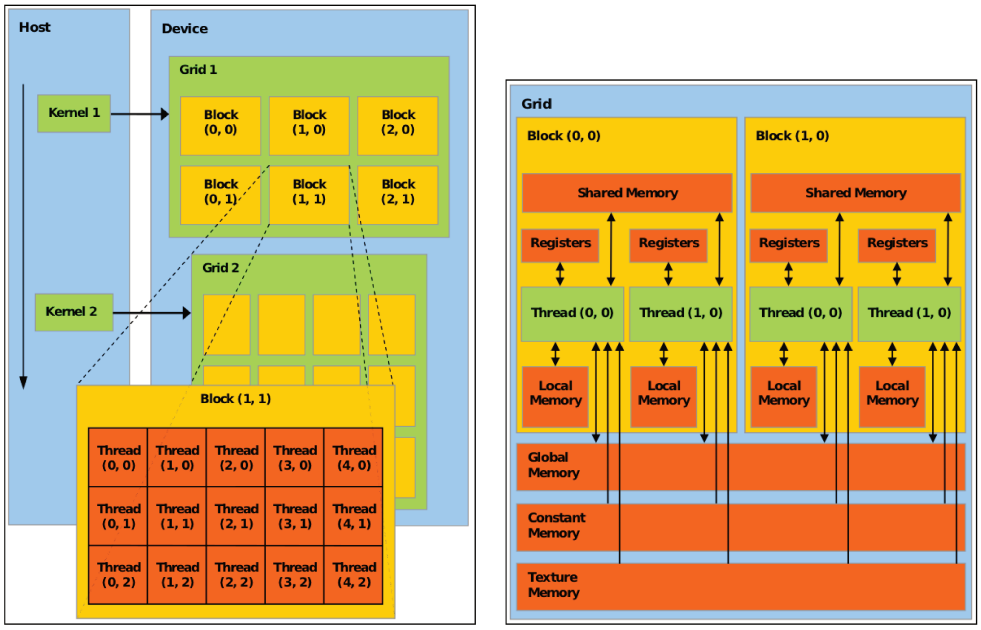


FIGURE 3.17 – A gauche : organisation des unités de traitement sur architecture CUDA : une tâche executée sur le GPU s'appelle un kernel. Celui-ci est réparti sur une grille (Grid) divisée en blocs (Block), eux-mêmes divisés en threads qui sont les unités atomiques de calcul.
A droite : répartition de la mémoire dans une grille : chaque bloc possède une mémoire locale. De façon classique, plus une mémoire est proche d'une unité de traitement, plus elle est véloce mais petite. (Nvidia)

Le GPGPU

Bien que conçu pour le jeu-vidéo et la synthèse d'image, en réalité rien n'empêche d'utiliser un GPU pour réaliser des calculs divers. C'est le principe du *General-purpose Processing on Graphics Processing Units*, abriévé GPGPU. Cette pratique n'est pas récente mais elle a connu une expension importante à partir du moment où les constructeurs de cartes graphiques ont donné la possibilité aux programmeurs d'intervenir sur la file de traitement par le biais de *shaders*. Avec un peu d'astuce, il était ainsi possible de réaliser des calculs divers et d'écrire le résultat dans une texture, chaque pixel étant une valeur calculée. Devant l'intérêt d'une telle utilisation de leurs matériels, les constructeurs ont élargi ces possibilités par le biais d'API dédiées dans un premier temps, puis de modifications de leur hardware pour prendre en compte cette utilisation : ce sont les architectures ATI STREAM et CUDA. Dans notre étude, nous avons surtout utilisé cette dernière que nous allons décrire sommairement. Toutefois les approches sont globalement similaires.

Pour faire simple, le code exécuté en une tâche sur un processeur s'appelle un kernel. Cette tâche est répartie sur une grille de calcul à M dimensions, divisée en blocs. Chaque bloc est à nouveau divisé en N *threads*, la « structure » atomique. Au sein d'un bloc, ces threads sont executés par groupes de 16 ou 32, selon l'architecture ; ces groupes sont appelés des *warps*. La figure 3.17 schématise cette architecture.

Implantation naïve

Au centre de l'algorithme CORE, se trouve un estimateur par noyaux. C'est cette étape qui pose problème dans les temps de calcul. Pour rappel, nous devons attribuer un critère

C pour chaque point d'intérêt i à l'aide de l'équation :

$$\mathrm{C_i} = \frac{1}{(\mathbf{N}-1)} \sum_{\mathbf{j} \neq \mathbf{i}} \mathrm{K}\left(\left|\mathbf{u_i} - \mathbf{u_j}\right|\right) \tag{3.45}$$

Pour chaque vecteur caractéristique, nous avons donc une somme des distances par rapport aux autres vecteurs à calculer. Ce processus étant indépendant des autres, nous pouvons le réaliser dans un *thread* CUDA.

Notre optimisation

Comme évoqué brièvement, si nous nous intéressons à nouveau à l'équation (3.45) il n'est pas difficile de réaliser que certains calculs se font en double. La symétrie de la distance fait qu'il est redondant de calculer $\left|\mathbf{u_i} - \mathbf{u_j}\right|$ et $\left|\mathbf{u_j} - \mathbf{u_i}\right|$. Nous pouvons donc mémoriser ces résultats dans une matrice de distances où seule une moitié est utilisée, comme l'illustre la figure 3.19. Ainsi, une version améliorée de l'algorithme sur GPU passerait par sa scission en deux étapes de calcul : la première étant celle de la matrice des distances, la seconde des critères C.

Afin d'éviter un gaspillage des emplacements dans la mémoire du GPU, il est préférable de stocker cette matrice sous forme de tableau uni-dimensionnel qui serait la liste de la moitié des éléments de celle-ci. Le problème suivant se pose alors : comment retrouver de façon efficace les coordonnées p, q (ligne et colonne) à partir de l'indice unidimensionnel i ? Si nous avons n vecteurs caractéristiques, alors la matrice est de taille n^2. Mais pour avoir la taille de sa forme raccourcie, il faut enlever la dimension de la diagonale puis diviser par deux, nous avons donc le nombre d'éléments suivant :

$$\frac{n^2 - n}{2} = \frac{n(n-1)}{2} \tag{3.46}$$

Immédiatement, nous constatons que si nous connaissons le numéro de ligne p, le numéro de colonne est trivial à calculer :

$$q = (i + \delta) \mathrm{mod}(n) \tag{3.47}$$

avec δ un décalage fonction du numéro de ligne p. Ce dernier, en revanche, est moins immédiat à obtenir. Pour y arriver, considérons la sous-moitié de matrice après la ligne p. Son nombre d'éléments est :

$$\frac{(n-p-1)(n-p)}{2} \tag{3.48}$$

Si nous retranchons le nombre total d'éléments au nombre de la sous-moitié, nous avons, en partant du principe que i est sur la dernière colonne :

$$\frac{n^2 - n}{n} - \frac{1}{2}(n^2 - np^2 + p^2 - n + p) \tag{3.49}$$

$$\frac{1}{2}(n^2 - n - n^2 + 2pn - p^2 + n - p) \tag{3.50}$$

$$p^2 + 2pn + p = 2i \tag{3.51}$$

Nous aboutissons ainsi à une équation du deuxième degré, très simple à résoudre :

$$p^2 + (1-2n)\,p + 2i = 0 \tag{3.52}$$

Avec le discriminant :

$$\Delta = (1-2n)^2 - 8i \tag{3.53}$$

La solution est donc :

$$\frac{2n-1-\sqrt{(1-2n)^2-8i}}{2} \tag{3.54}$$

Reste maintenant à parcourir la matrice des distances pour calculer un critère. Un parcours type est illustré par la figure 3.19, ce qui nous donne le pseudo-code d'un *kernel* avec l'algorithme 4. Les résultats des temps de calculs sont présentés avec la figure 3.18.

Algorithme 4 : Kernel optimisé du calcul d'un critère

Données : id ← Indice du kernel (et du critère)
Données : N ← Nombre de vecteurs
Données : D ← tableau des distances
Données : h ← paramètre fenêtre de l'estimateur
Données : K ← dimension des vecteurs
Résultat : c ← critère calculé
$c \leftarrow 0$ $x \leftarrow 0$
$y \leftarrow$ id
$\delta \leftarrow 0$
$\epsilon \leftarrow$ N − 1
tant que $x < y$ **faire**
  $i \leftarrow \delta + y$
  $\epsilon \leftarrow \epsilon - 1$
  $\delta \leftarrow \delta + \epsilon$
  $d \leftarrow$ D[i]
  $c \leftarrow$ c + Kernel(d, h)
  $x \leftarrow x + 1$
fin
$y \leftarrow y + 1$
tant que $y <$ N **faire**
  $i \leftarrow \delta + y$
  $y \leftarrow y + 1$
  $d \leftarrow$ D[i]
  $c \leftarrow$ c + Kernel(d, h)
fin
$c \leftarrow \frac{c}{(\mathrm{N}-1)h\sqrt{2\pi}^{\mathrm{K}}}$
retourner c

3.4.2 CPU

Non véritablement étudié ici, nous proposons malgré tout des pistes d'améliorations sur un processeur classique. La première, évidente, reprend l'idée qui a guidé notre algorithme sur GPU : nous pouvons très bien calculer les distances une seule fois pour les

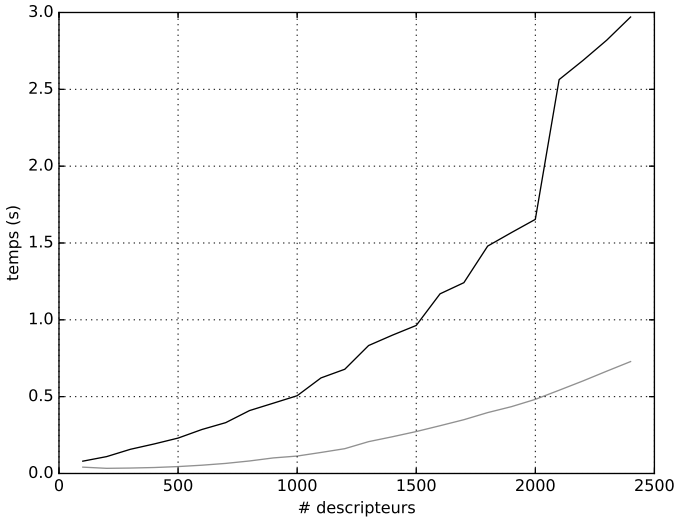


FIGURE 3.18 – Comparaison des temps de calcul en seconde de notre optimisation GPU (en gris) et d'une implantation naïve (en noir) en faisant varier le nombre de descripteurs de 100 à 2500, pour une dimension fixée de 128.

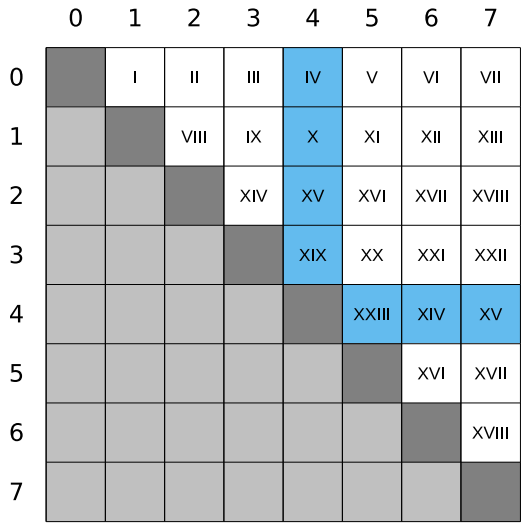


FIGURE 3.19 – Matrice des distances pour huit vecteurs caractéristiques. Seule la moitié de la matrice est utilisée. En bleu, un exemple de parcours pour calculer un critère (celui du vecteur d'indice 4).

mémoriser par la suite. Nous ne nous y attardons pas plus mais nous proposons des perspectives d'améliorations plus poussées dans notre conclusion.

3.5 Conclusions

Dans ce chapitre nous avons mis en lumière le problème de la confusion des descripteurs de points d'intérêt. Nous avons montré comment ce problème existe dans la littérature indépendamment du contexte de traitement d'images de documents, mais aussi pourquoi ce dernier est en réalité particulièrement concerné. Ceci vient, entre autres, de l'inadéquation des détecteurs de points d'intérêts et des descripteurs associés aux caractéristiques particulières des images de document qui exacerbent ces problèmes.

Dans un souci de réaliser le pont entre la communauté « Vision » et la communauté « Document » nous nous sommes intéressés aux méthodes « Vision » pouvant être appliquées dans un contexte d'images de documents avant de développer notre propre solution. Celle-ci, l'algorithme CORE, a été pensée pour être le plus simple possible d'utilisation, sans contraintes afin de faciler le pont. C'est un filtre générique qui repose un calcul de probabilité de confusion des descripteurs, ne nécessitant ni entrainement, ni apprentissage. Un autre atout majeur est qu'il fonctionne avec des vecteurs constitués de nombres flottants mais aussi de bits, pouvant ainsi être utilisé avec des descripteurs binaires qui sont particulièrement en vogue ces dernières années. Nous avons testé notre proposition sur l'estimation d'une homographie entre deux images, à la fois dans un contexte de documents mais aussi sur des images plus génériques ainsi que dans une application de segmentation de document dans un flux vidéo sur le jeu de données d'envergure internationale, SmartDOC. Les résultats sont tout à fait satisfaisants et valident la pertinence de notre approche.

Conscients des enjeux de la capture nomade à notre époque, nous nous sommes aussi intéressés au temps de calcul nécessaire pour utiliser l'algorithme CORE. A ce sujet, nous avons fait des propositions pour réduire la charge d'utilisation des processeurs, autant sur architecture CPU que GPU. Ces propositions dépassent d'ailleurs le cadre de notre étude et peuvent être utilisées dans des contextes et applications diverses et variées, à partir du moment où il est nécessaire de calculer des distances entre des ensembles de vecteurs.

Pour conclure, nous sommes en mesure d'affirmer que ce travail a abouti de manière satisfaisante, conformément aux objectifs qui étaient les notres ; l'algorithme CORE est facile d'utilisation et performant, il est générique et peut s'appliquer dans n'importe quel contexte utilisant des descripteurs de points d'intérêt. Toutefois, ce souci de généricité nous a quelque part limité ; après tout une image de document comporte des caractéristiques très particulières et il est regrettable de ne pas avoir cherché à les exploiter directement. Cette réflexion est la base du chapitre qui suit.

Chapitre 4

Détection et document

« *Administrative - "Que voulez-vous ?"*
Astérix - "Le laissez-passer A-38."
Administrative - "Vous avez le formulaire bleu ?"
Astérix - "Le formulaire bleu ? Non."
Administrative - "Alors comment voulez-vous obtenir le laissez-passer A-38 ?" »

Les Douze Travaux d'Astérix - 1976

Sommaire

4.1 Introduction

Dans le chapitre 2 nous avons modestement retracé l'historique de la construction des détecteurs et descripteurs de points d'intérêt en essayant de nous attarder sur le tableau global de leur évolution. Il nous est apparu logiquement que ceux-ci étaient relativement inadaptés quant à une utilisation dans le cadre d'une application impliquant des images de documents. Rusiñol *et al.* ont réalisé en 2015 une étude comparative très riche en enseignements sur la question pour une application à la classification de documents [Rusiñol et al., 2015].
Nous avons alors formulé dans le chapitre 3 une proposition visant à pallier ces manquements en attaquant ce problème par le biais de la notion de confusion des descripteurs ; il s'agit de notre algorithme CORE. L'un de ses atouts principaux est de ne pas perturber les processus classiques de visions par ordinateur afin d'y faciliter son intégration.

En mettant de côté ce souci de généricité, il est bien évidemment possible d'imaginer d'autres méthodes qui prennent en compte dès le départ la spécificité des images de documents qui possèdent des caractéristiques propres que nous pouvons utiliser comme information à priori. Cette réflexion est l'objet de ce chapitre.

4.1.1 Points d'intérêt dédiés

Certains détecteurs sont conçus pour ne fonctionner qu'avec des images de documents. Nous passons en revue quelques réalisations.

SITT

SITT est une tentative de définition d'un extracteur de features dédié aux images de documents, proposé par Block *et al.* en 2007 [Block et al., 2007]. Le but recherché était l'amélioration des méthodes d'OCR après construction d'une mosaïque d'images avec RANSAC. Les auteurs, ayant eux aussi constaté le très grand nombre de réponses retournées par les algorithmes basés sur des calculs de gradients, proposent plutôt de rechercher directement dans l'image binarisée des motifs qu'ils estiment discriminants, comme par exemple des signes de ponctuations.

Toutefois, l'inconvénient d'une simple recherche de motif est un grand nombre de réponses retournées lorsque l'image est bruitée ou lorsque des éléments d'une scène y sont présents ; que ce soit par l'inclusion d'une photographie dans le document ou la capture d'éléments extérieurs. Pour y remédier, les auteurs proposent une étape de post-traitement basée sur la densité du voisinage des points détectés, ayant constaté que les faux positifs avaient tendance à être particulièrement nombreux dans une zone restreinte.

De l'aveu des auteurs, SITT est incomplet. Les résultats sont au mieux similaires à ceux obtenus par SIFT, ce qui pourrait être suffisant mais plusieurs résultats défavorables sont à souligner ce qui limite l'intérêt de la méthode. En revanche le rapport résultat sur temps de calcul est à leur avantage, ce qui est encourageant pour explorer cette idée. Ils proposent d'ailleurs quelques pistes dans leur article, comme l'identification des lignes pour permettre de calculer une distance relative des points par rapport à celles-ci afin d'ajouter plus d'information discriminante. Malheureusement il semble que ces travaux n'aient pas été poursuivis par la suite.

LLAH

Likely Locally Arrangement Hashing (LLAH) est une méthode relativement récente d'indexation et de recherche d'images de documents à partir d'acquisitions par un appareil photo [Nakai et al., 2006]. Cet algorithme répond avec brio à un problème réputé difficile en raison de deux éléments épineux que sont : la modification de la perspective de l'image à cause de la capture *nomade* (au contraire d'un scanner, il est peu concevable de réussir à reprendre le même objet en photo avec exactement le même angle de capture) et bien entendu la mise en correspondance efficace d'images de documents, avec toutes les particularités qu'elles comportent.

La solution retenue utilise une méthode de hachage géométrique des points d'intérêts de l'image. Originellement, il s'agit d'une approche très couteuse en temps de calcul ($O(N^5)$) et la contribution des auteurs de LLAH est particulièrement appréciée puisque la complexité de leur algorithme est linéaire au nombre de points. Pour garantir l'invariance par le changement de perspective, le calcul du cross-ratio est utilisé :

$$\frac{P(A,B,C)P(A,D,E)}{P(A,B,D)P(A,C,E)} \tag{4.1}$$

où $P(A,B,C)$ représente l'aire du triangle ABC. Par la suite, celui-ci est écarté au profit d'un invariant affine :

$$\frac{P(A,C,D)}{P(A,B,C)} \tag{4.2}$$

Les valeurs sont ensuite discrétisées selon leur fréquence d'apparition à l'aide d'un histogramme des valeurs construit au préalable et proposé par les auteurs à partir de leurs expérimentations.

Enfin, et le plus important pour notre étude, l'extraction des points d'intérêts est très simple et se fait en quatre étapes. Tout d'abord, une première binarisation de l'image est calculée à l'aide d'un seuillage adaptatif. Le résultat est flouté par un filtre gaussien dont les paramètres sont déterminés en fonction de l'estimation de la taille des caractères du document (la racine carrée de la valeur dominante des surfaces des composantes connexes de l'image binarisée). Un deuxième seuillage adaptatif est appliqué et enfin les points d'intérêt sont extraits à partir des centroïds des composantes connexes, ce qui correspond globalement aux mots du texte.
L'idée de prendre les centroïds des composantes connexes formant les mots est efficace dans le domaine de l'indexation et la recherche d'images de documents. Elle est d'ailleurs reprise récemment dans le descripteur SRIF [Dang et al., 2015], une amélioration de LLAH ou dans [Dang et al., 2016] lors de l'élaboration du descripteur DETRIF (*DElaunay-based TRIangulation Features*) basé, comme le nom l'indique, sur une triangulation de Delaunay [Delaunay, 1934].

Les mérites de LLAH sont nombreux : la complexité algorithme est faible et permet un enregistrement et une recherche dans une base de données très rapidement. De plus, son efficacité est remarquable et l'approche d'extraction des points d'intérêt à partir des centroïds des mots ne nécessite pas des images de résolution élevée, la capture pouvant se faire avec une simple *webcam* en temps réel. Toutefois, l'inconvénient majeur de cette méthode est de ne fonctionner qu'avec principalement des documents ne comportant que du texte, les images ayant tendance à fausser l'extraction des points d'intérêt.

DTMSER

Distance Transform Based MSER (DTMSER) [Gao et al., 2013] sort un peu du cadre de notre étude car il ne s'agit pas d'un algorithme d'extraction de points d'intérêt mais de région d'intérêt ; l'emplacement détecté étant une zone et non un pixel. Toutefois il nous parait important de le mentionner puisqu'il a été conçu pour des images de documents et qu'il présente le double avantage d'être simple et efficace. De plus, il se base sur une approche que nous jugeons intéressante qui est l'analyse de la structure du document.

L'algorithme MSER (*Maximally stable extremal regions*) [Matas et al., 2002] d'origine détecte les régions qui possèdent un changement d'intensité fort par rapport à leur environnement immédiat. Résumons rapidement son fonctionnement : l'image est d'abord convertie en niveau de gris et des seuillages globaux successifs sont appliqués en faisant varier la valeur du seuil. Pour chaque étape, l'ensemble des composantes connexes possibles sont extraites. Parmi celles-ci, il y en a certaines qui sont présentes lors de différentes valeurs de seuillage : ce sont des régions dites « stables » et elles possèdent des caractéristiques recherchées comme l'invariance aux transformations ou la détection selon différentes échelles.

Sur les images de documents, il est possible d'avoir un résultat qui soit équivalent à une analyse en composantes connexes. Les auteurs de DTMSER ayant observé que cette approche était efficace pour la problématique de recherche d'images appliquée aux images de comics, ils proposent une extension pour les images de documents textes en l'appliquant sur l'image de la transformée de la distance. Celle-ci est définie pour chaque pixel p de l'image par l'équation suivante :

$$f(p) = \min_{q \in Q} d(p,q) \tag{4.3}$$

avec $d(p,q)$ la distance euclidienne entre p et un objet q. L'application de MSER sur l'image de la transformée de la distance produit un dendogramme des zones de variations. Ce dernier est par ailleurs riche en information puisqu'il représente les relations spatiales entre les différentes régions.

DTMSER est très rapide. C'est un argument majeur mais ce n'est heureusement pas le seul. Il est capable de détecter des régions qui ont un sens sémantique dans le document, en raison de leur importance qu'on leur accorde dans le texte là où des critères visuels habituels perdent une certaine disciminabilité. D'ailleurs, le nombre de réponses retournées est beaucoup plus faible que les détecteurs classiques, ce qui n'empêche pas d'après les auteurs d'avoir des performances légèrement supérieures à ces approches, preuve de la pertinence de cette méthode. D'après [Gao et al., 2014], cet algorithme se prête d'ailleurs très bien aux approches *bag-of-words*.

4.1.2 Analyse de la structure

La structure d'un document texte n'est pas issue du hasard, elle a été construite dans le but de fluidifier la lecture. Nous partons donc du principe qu'il s'agit d'une information visuelle trop importante pour être négligée et nous nous intéressons aux méthodes d'extraction de l'agencement du document, le *layout* en anglais. Nous pouvons voir ce processus comme la transformation d'une représentation d'une image de document par

FIGURE 4.1 – Les différentes catégories de disposition de documents classiquement rencontrées, de gauche à droite : *manhattan*, *non-manhattan* (Copyright (c) 2012. EPITA Research and Development Laboratory (LRDE) with permission from Le Nouvel Observateur), *divers*.

pixels en une représentation de plus haut niveau par formation de lignes, blocs de lignes et éléments graphiques.

La manière d'aborder ce problème change radicalement selon le type de document à traiter, ou comment celui-ci est agencé. Nous avons par exemple les dispositions dites de « *Manhattan* » en référence à la célèbre ville organisée en blocs de quartiers rectangulaires. Les articles scientifiques sont reposent souvent sur une telle disposition, sans fioritures. Ce type d'agencement est le plus simple à traiter, les premières méthodes se sont par conséquent focalisées dessus. Viennent ensuite les dispositions « *non-Manhattan* », qui peuvent être un peu moins géometriques mais restent relativement bien ordonnées : ce sont par exemple ce que l'on observe dans les magazines. La dernière catégorie où sont regroupées toutes celles qui ne rentrent pas dans les deux premières est souvent appelée « *chevauchement* » : agencements sans véritables règles où il est difficile voire impossible de délimiter clairement des zones rectangulaires ; documents manuscrits ou historiques sont typiquement dans cette catégorie. La figure 4.1 illustre ces différentes dispositions de documents.

Pour répondre à ces différentes classes de documents, il existe différentes catégories de méthodes. Nous résumons les deux plus importantes qui sont dans l'ordre d'apparition « *top-down* » et « *bottom-up* ». La première catégorie concerne les méthodes qui fonctionnent par fusion de petits éléments du document jusqu'à arriver progressivement à l'obtention des zones principales. La deuxième catégorie concentre les méthodes qui procèdent par découpements récursifs du document jusqu'à arriver au seuil souhaité.

Dans notre étude nous ne présentons pas d'état de l'art complet sur la question. Celle-ci est très riche, nous intéresser plus en profondeur au problème nous aurait fait manquer de temps par rapport aux autres travaux sur le filtrage des points d'intérêt. Nous discutons malgré tout de quelques méthodes emblématiques afin de présenter un tour d'horizon relativement complet des approches que nous pouvons rencontrer dans la littérature.

Bottom-up

Les méthodes *Bottom-up* sont historiquement les premières à être apparues dans la littérature, il est donc normal que nous commencions notre revue par celles-ci. Elles ont en plus l'avantage d'être souvent plus intuitives que les méthodes *Top-down*.

Une première catégorie d'algorithmes sont les méthodes de *smearing* fonctionnant par étalement des composantes connexes. RLS [Wong et al., 1982], pour *Run-Lentgh-Smearing* est un algorithme emblématique sur ce principe qui a inspiré de nombreuses variantes. Visuellement, cette méthode donne l'impression « d'étaler » l'encre sur le document dans une direction donnée. L'image est d'abord binarisée, les pixels blancs étant représentés par des 0 et les noirs par des 1. Ensuite, chaque séquence binaire x, extraites à partir de chaque ligne de l'image, est convertie en une séquence binaire y selon les règles suivantes :

1. les pixels à 1 dans x le restent dans y,
2. les pixels à 0 dans x prennent la valeur 1 dans y si le nombre de pixels adjacents valant 0 dans x est inférieur ou égal à un seuil T prédéfini.

L'opération est ensuite répétée séparement sur les séquences binaires verticales de l'image et les deux masques résultats sont combinés avec un opérateur logique ET pour obtenir le masque final de l'agencement du document, après quelques opérations de nettoyage pour eviter des faux positifs évidents. Les étapes de l'algorithme sont illustrées par la figure 4.2.

Dans des conditions maitrisées, cet algorithme est très efficace. En revanche sa simplicité impose l'utilisation d'images parfaitement droite ou alors d'avoir recours à une correction de l'orientation, en plus de ne s'occuper que des agencement *manhattan*. Il en va de même pour la qualité de l'image d'origine, l'opération de binarisation devant être sans reproche. [Hinds et al., 1990] propose quelques améliorations avec des paramètres dynamiques ainsi que [Okamoto and Takahashi, 1993] qui rajoute un peu plus d'analyses empiriques sur l'organisation du document.

Docstrum [O'Gorman, 1993] est une autre méthode qui fonctionne sur des documents non-manhattan, par regroupement des composantes connexes à l'aide de l'algorithme des k plus proches voisins après une première étape de binarisation et de débruitage pour enlever les résidus. En fonction d'une analyse de la taille des caractères, les composantes connexes sont séparées en deux groupes qui sont : texte normal et titres & entêtes, cette dernière catégorie ayant une information très particulière. Les lignes de texte sont construites par clôture transitive avec seuillage, ce qui permet ensuite de mettre en évidence les paragraphes si les lignes sont parallèles et qu'elles respectent un seuil de distance pré-défini. Enfin, un intérêt appréciable de cet algorithme est qu'il permet en même temps d'estimer l'orientation du document. [Strouthopoulos et al., 1997] propose une amélioration quatre ans plus tard qui amène de meilleurs résultats simplement en utilisant des boites englobantes plutôt que le centroïd pour chaque composante connexe.

Kise *et al.* [Kise et al., 1998] propose une segmentation par Voronoï à partir d'un échantillon de points des contours des composantes connexes de l'image, après binarisation et débruitage. Une série d'analyses empiriques assure la suppression des faux positifs en fonction de la taille et de l'emplacement. [Agrawal and Doermann, 2009] propose une améloriation en combinant cette méthode avec l'algorithme docstrum.

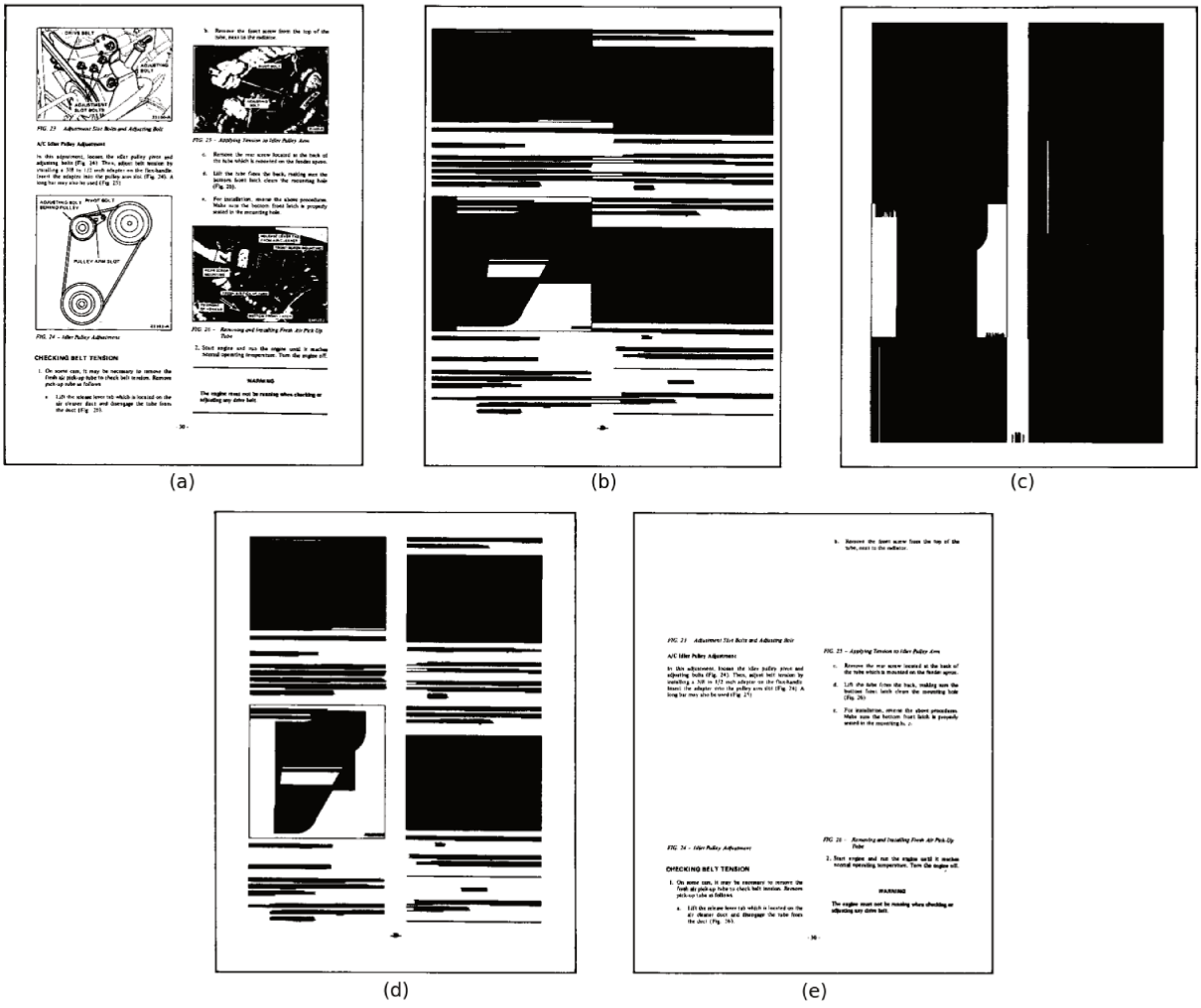


FIGURE 4.2 – Segmentation par *Run-Lentgh-Smearing*. (a) image originale (b) RLS horizontal (c) RLS vertical (d) combinaison des deux masques (e) extraction du texte. [Wong et al., 1982]

Enfin, comme bien souvent il est possible d'avoir recours à des méthodes s'appuyant sur un apprentissage. C'est le cas de [Nicolas et al., 2006] qui réalise une application des champs de Markov [Kindermann and Snell, 1980] pour la détection des zones de texte dans une approche qui ressemble beaucoup à de la binarisation en déterminant la classe X en fonction de la valeur du pixel Y : il s'agit donc d'estimer la loi de probabilité P(Y/X). L'auteur utilise l'algorithme EM pour déterminer la caractérisation de cette loi à partir de vecteurs caractéristiques construit avec les valeurs des pixels locaux. L'inconvénient d'une telle approche est toutefois la nécessité d'être appliquée sur des documents de même nature, en raison de l'apprentissage.

Top-down

Parfois un peu plus subtiles que les méthodes *bottom-up*, nous présentons maintenant quelques contributions emblématiques *top-down*.

XY-Cut est le premier algorithme de cette catégorie a avoir été conçu [Nagy, 1984]. Plutôt que de chercher l'emplacement des zones de texte, l'idée est au contraire de détecter les espaces inter-lignes. Ce processus repose sur la construction de projections de l'image qui sont des sommations des valeurs de pixels en ligne et en colonne : à partir de ces profils, il est facile de détecter les espaces interlignes qui correspondent à des creux. Pour chaque endroit où un creux sur la projection horizontale a été détecté, une projection verticale est appliquée. Ce processus est répété jusqu'à ce qu'il n'y ait plus de creux détecté ou que les blocs retournés soient inférieurs à une taille T. Un avantage intéressant de cet algorithme, lié à son fonctionnement récursif, est qu'il permet de construire un arbre donnant l'information sur quel élément en contient d'autres ; c'est une information riche, [Cesarini et al., 2001] d'ailleurs est une modification de XY pour améliorer encore plus cette information hiérarchique. En revanche l'utilisation de la projection implique d'utiliser cet algorithme sur des documents de type *manhattan* exclusivement et nécessite des images sans orientations. [Wieser and Pinz, 1993] Propose une amélioration en combinant avant RLS. Cet algorithme emblématique reste utilisé, comme très récemment Corbelli *et al.* qui s'appuient dessus comme première étape de leur segmentation de documents historiques [Corbelli et al., 2016].

Dans le même principe (recherche des emplacements ne contenant pas de texte), [Baird, 1992] est à l'origine d'une contribution fonctionnant par regroupement des recouvrements de rectangles qui représentent le fond du document. Breuel propose d'ailleurs quelques années plus tard une version améliorée de cette méthode [Breuel, 2002]. Une fois les rectangles détectés, ceux-ci sont triés selon un indice K :

$$\mathrm{K}(c) = \sqrt{\mathrm{surface}(c) \times \mathrm{W}(|\log_2(\mathrm{H}(c)/\mathrm{L}(c))|)} \tag{4.4}$$

avec W une fonction de pondération afin de donner plus de poids aux séparateurs de paragraphes, où H et L correspondent respectivement à la hauteur et à la longueur du rectangle c.

$$\mathrm{W}(x) = \begin{cases} 0.5 \text{ si } x < 3 \\ 1.5 \text{ si } 3 \leqslant x < 5 \\ 1 \text{ sinon} \end{cases} \tag{4.5}$$

Les rectangles c sont ensuite ajoutés progressivement à un accumulateur S, qui représente le recouvrement total du document, jusqu'à que l'inégalité suivante soit satisfaite :

$$\mathrm{K}(s) - \mathrm{W}_s \times \mathrm{F(S)} \leqslant \mathrm{T}_s \tag{4.6}$$

Avec K(s) l'indice K du dernier rectangle ajouté, F(s) = $\frac{j}{m}$, W_s un coefficient de pondération et T_s un seuil empirique. A la fin, les boites englobantes correspondant aux zones non recouvertes par l'union S sont les zones contenant du texte.

Enfin, souvent négligée dans notre domaine, certaines approches ont recours à l'utilisation de l'information liée à la couleur. C'est le cas de [Kim, 1996] et [JAIN and YU, 1998] qui reposent sur l'analyse des histogrammes par couleurs pour aider à segmenter l'image, finissant le travail par un jeu d'heuristiques classiques.

4.2 Binarisation

Le traitement automatisé de l'information pour une image de document se heurte souvent, comme première difficulté, à la représentation de l'image elle-même : c'est un objet complexe. Nous percevons le monde en couleur, aussi les capteurs que nous fabriquons aujourd'hui en font de même mais l'information qui est rajoutée complique la tâche. C'est pour cela que la plupart du temps, avant toute opération, l'image est convertie en niveaux de gris où chaque pixel représente une intensité lumineuse.

Cette première étape est indispensable mais pas suffisante pour de nombreuses applications qui nécessitent d'avoir clairement le texte séparé du fond. Les valeurs possibles des pixels sont réduites, du traditionnel $0-255$, à deux valeurs : 0 et 1. Ceci est appelé la binarisation et c'est un sujet d'importance, bien moins simple qu'il en a l'air.

4.2.1 Seuillage global

Les méthodes de binarisation les plus simples sont celles du seuillage global. Tous les pixels en dessus -ou en dessous- d'une certaine intensité T sont crédités de la valeur 1, 0 sinon. La transformation de binarisation B d'une image I peut s'écrire ainsi :

$$\forall p \in \mathrm{I},\ \mathrm{B}(p) = \begin{cases} 1 \text{ si } \mathrm{I}(p) > \mathrm{T} \\ 0 \text{ sinon} \end{cases} \tag{4.7}$$

Dans des environnements totalement maitrisés où les documents employés sont tous calibrés, le seuil T est choisi empiriquement. Mais cela est rarement le cas, et c'est pourquoi les seuillages globaux reposent le plus souvent sur l'analyse de l'histogramme des valeurs de gris de l'image pour trouver la bonne séparation.

Un des algorithmes les plus célèbres est celui de la méthode d'Otsu [Otsu, 1979]. En partant du principe que l'histogramme est bi-modale, nous cherchons un seuil t qui minimise la variance intra-classe pondérée :

$$\sigma_w^2 = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{4.8}$$

avec q la probabilité d'occurrence d'une classe, obtenue à partir de l'étude de l'histogramme normalisé comme sommation des probabilités pour chaque valeur de pixel de la classe :

$$q_1(t) = \sum_{i=1}^{t} \mathrm{P}(i) \quad q_2(t) = \sum_{i=t+1}^{\mathrm{I}} \mathrm{P}(i) \tag{4.9}$$

Les moyennes de chaque classe sont définies par :

$$\mu_1(t) = \sum_{i=1}^{t} \frac{i\mathrm{P}(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^{\mathrm{I}} \frac{i\mathrm{P}(i)}{q_2(t)} \tag{4.10}$$

ce qui nous permet de calculer les variances de chaque classe :

$$\sigma_1^2 = \sum_{i=1}^{t} \left[i - \mu_1(t)\right]^2 \frac{\mathrm{P}(i)}{q_1(t)} \qquad \sigma_2^2 = \sum_{i=t+1}^{\mathrm{I}} \left[i - \mu_2(t)\right]^2 \frac{\mathrm{P}(i)}{q_2(t)} \tag{4.11}$$

Il est tout à fait possible de s'arrêter là : on évalue pour chaque valeur t possible la variance intra-classe $\sigma_w^2(t)$ et sélectionne la plus petite. Mais Otsu propose une accélération basée sur l'expression de la variance totale :

$$\sigma^2 = \sigma_w^2(t) + \sigma_\mathrm{B}^2(t) \tag{4.12}$$

avec $\sigma_\mathrm{B}^2(t)$ la variance inter-classes définie par :

$$\sigma_\mathrm{B}^2(t) = q_1(t)\left[1 - q_1(t)\right]\left[\mu_1(t) - \mu_2(t)\right]^2 \tag{4.13}$$

Par conséquent, minimiser la variance intra-classe revient à maximiser la variance inter-classes. Or, cette dernière est plus rapide à calculer par récursion, $q(t+1)$ étant obtenu par $q(t) + \mathrm{P}(t+1)$ et de même pour μ_1 et μ_2.

4.2.2 Seuillage local

Malheureusement, le « monde idéal » où le seuillage global peut s'appliquer est loin d'être la norme : bien souvent, une telle situation n'existe pas. Par exemple, si le document est mal illuminé (et c'est très souvent le cas lors d'une capture nomade) le seuil idéal T va varier selon l'emplacement dans l'image. Aussi les méthodes locales vont analyser l'environnement de chaque pixel afin de trouver le seuil le plus adapté.

Niblack [Niblack, 1985] est à l'origine d'une contribution illustrant simplement cette idée, par une analyse statistique locale. Le seuil T est calculé à l'emplacement x et y suivant l'équation :

$$\mathrm{T}(x,y) = \mu(x,y) + k\sigma(x,y) \tag{4.14}$$

où μ représente la moyenne des pixels locaux et σ l'écart type, avec k un coefficient de pondération choisi empiriquement autour de 0.2. Cette approche est simple mais présente le défaut de n'être que peu résistante au bruit : des pixels de forte intensité vont faire augmenter l'écart type σ artificiellement. Pour contourner ce problème, Sauvola propose une amélioration qui est plus utilisée [Sauvola et al., 1997] en introduisant une constante R afin d'ajuster la dynamique de σ :

$$\mathrm{T}(x,y) = \mu(x,y) + k\left(\frac{\sigma(x,y)}{\mathrm{R}} - 1\right) \tag{4.15}$$

[Shafait et al., 2008] propose l'utilisation d'images intégrales pour calculer très rapidement le seuil local d'une image g.

$$\mathrm{I}(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} g(i,j) \tag{4.16}$$

Cela permet d'obtenir la moyenne m et la variance s^2 en un temps linéaire :

$$m(x,y) = \frac{(\mathrm{I}(x+w/2,y+w/2)+\mathrm{I}(x-w/2,y-w/2)-\mathrm{I}(x+w/2,y-w/2)-\mathrm{I}(x-w/2,y+w/2)}{w^2} \tag{4.17}$$

$$s^2(x,y) = \frac{1}{w^2}\sum_{i=x-w/2}^{x+w/2}\sum_{j=y-w/2}^{y+w/2} g^2(i,j) - m^2(x,y) \tag{4.18}$$

En temps de calcul, une telle astuce permet d'avoir des résultats presque aussi rapidement qu'une méthode par seuillage global.

Enfin, nous pouvons citer [Ramírez-Ortegón et al., 2010] et [Block and Rojas, 2009] qui ont recours à l'analyse des pixels situés sur les bords des contours (un peu comme pour les méthodes par contour que nous évoquons juste après), qu'ils nomment *transition pixels* pour déterminer le seuillage local à utiliser.

4.2.3 Autres approches

Plutôt que de percevoir le problème comme celui du traitement d'un signal ou de l'étude de la statistique d'une distribution, il est possible de l'aborder sous des angles plus « géométriques ».

Par exemple, Kim *et al.* en 2002 [Kim et al., 2002] est à l'origine d'une contribution originale dite de « montée des eaux ». L'image est représentée comme un paysage dont la topographie est représentée par l'intensité des valeurs de gris des pixels ; c'est une surface 3D. Une simulation de pluie uniforme sur ce paysage provoque un écoulement vers les zones en vallées par recherche locale du plus bas niveau de gris. Après un certain nombre d'itérations, l'image est segmentée en deux classes : les zones sèches et les zones humides. L'inconvénient de cet algorithme est son critère d'arrêt : le nombre d'itérations est fixé à l'avance, ce qui manque un peu d'adaptabilité selon les images à traiter.

Puisque l'objectif après tout est bien souvent d'extraire le texte dans le document, certaines méthodes ne se focalisent pas sur l'évaluation individuelle de chaque pixel en fonction de sa valeur mais sur l'étude des contours dans l'image. Dans ce domaine, nous pouvons citer l'algorithme de Canny [Canny, 1986] qui est un incontournable. Le problème qu'il se pose le plus fréquemment est la fermeture des contours : il peut arriver que l'algorithme de détection retourne un faux-négatif, entrainant une ouverture dans un contour. C'est le cœur des préocupations de ces méthodes. [Cao et al., 2000] propose d'utiliser l'information liée à l'orientation du contour pour pouvoir le fermer par prolongation et [Chen et al., 2008] fait de même en rajoutant une information de distance par rapport aux autres contours afin d'éviter les faux-positifs.

Enfin, et nous sommes sensibles à ces approches, certaines contributions se penchent vers l'emploi de la théorie de la probabilité. C'est le cas de [Wolf and Doermann, 2002] qui utilise les champs de markov pour binariser des documents. Puisque il y a une notion de voisinage dans ce modèle stochastique, c'est aussi une forme de seuillage local qui s'accommode bien au bruit. De telles contributions sont souvent efficaces, comme l'algorithme FAIR [Lelore and Bouchara, 2013] qui a remporté le concours DIBCO en 2011 [Pratikakis et al., 2011].

Il existe bien d'autres approches. Des combinaisons d'algorithmes comme [Gatos et al., 2006], des segmentations en plusieurs classes avant de réduire à deux [Fabrizio et al., 2009], [Trier and Taxt, 1995] Nous ne pouvons pas toutes les citer sachant que notre étude ne s'est pas concentrée sur le sujet. La littérature dans ce domaine est riche. Ce problème est fortement dépendant du contexte et du type d'images à traiter : fond non-uniforme, éclairage non-uniforme, document dégradé, couleurs... Il est souvent nécessaire d'adapter les algorithmes au problème étudié.

4.2.4 Contribution

Étant une des premières étapes du domaine, la binarisation a naturellement inspiré nos premières réflexions en guise d'introduction au sujet ce qui a abouti à la très humble proposition qui suit. Celle-ci trouve son origine dans les outils apportés par les automates cellulaires après la découverte d'une nouvelle méthode de calcul.
Mais avant d'aller plus loin, il est nécessaire de rappeler au lecteur ce que sont les automates cellulaires. Stephen Wolfram, auteur d'un ouvrage remarquable sur la question, utilise la définition suivante : *les automates cellulaires sont des idéalisations mathématiques de systèmes physiques dans lesquels l'espace et le temps sont discretisés et où les quantités physiques sont représentées par un ensemble fini de valeurs discrètes.* Ainsi, il est parfaitement possible de réaliser une simulation d'écoulement de fluide selon les équations de Navier-Stroke à l'aide d'un automate cellulaire. Mais plus généralement, une définition formelle mathématique représente un automate cellulaire comme étant un 4-uplet (E, N, V, s) où :

1. E est un réseau, au sens spatial,
2. N est l'ensemble des états possibles,
3. V est une fonction de voisinage,
4. s est une fonction de transition.

Le réseau étant une structure géométrique discrète, nous pouvons manipuler des structures à D dimensions. On parle alors d'automates à D dimensions. Toutefois les automates bidimensionnels sont les plus fréquents, particulièrement dans notre champ d'étude où l'image constitue directement le réseau. Nous pouvons citer quelques contributions marquantes : détection du texte dans une scène urbaine [Zagoris and Pratikakis, 2012], transformations artistiques d'images [Petric,], détection de régions saillantes [Jones et al., 2010] ou encore un ensemble de traitements simples (débruitage, enveloppe convexe, amincissement) par [Rosin, 2005].

Bien que ne présentant pas d'application immédiate, l'un des automates cellulaires le plus célèbre est celui du jeu de la vie proposé par H. Conway en 1960. Comportant seulement deux états (mort ou vivant), les règles de sa fonction de transition sont tout aussi simples ce qui en a fait un sujet d'étude fréquent dans les formations informatiques : une

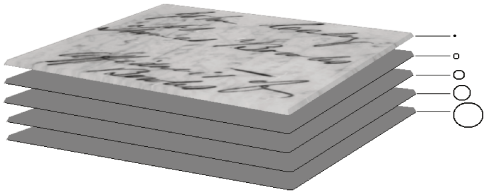


FIGURE 4.3 – Illustration d'un « cube de données » : chaque niveau est l'application du calcul du taux de remplissage pour une dimension de cercle donnée.

cellule morte passe à l'état vivant si à l'instant T elle possède exactement trois voisins vivants et une cellule vivante le reste si à l'instant T elle possède exactement deux ou trois voisins vivants. De plus, la simplicité de cet automate cache une richesse fascinante par l'apparition de plusieurs comportements émergents (effet global complexe provoqué par l'application de règles locales et simples), ce qui a inspiré des réflexions en théorie de la calculabilité et décidabilité. Notons pour terminer qu'il est possible de réaliser entièrement un automate simulant une machine de Turing. En 2011, [Rafler, 2011] a proposé une modification radicale en définissant *Smooth Life*, une version continue du célèbre jeu de la vie. Plutôt que de compter le nombre de voisins, l'auteur propose de calculer le taux de remplissage d'une zone qui définit la cellule ou un voisinage. Ce taux de remplissage est simplement obtenu par l'équation suivante :

$$\mathrm{M}(x,t) = \frac{1}{\pi h^2} \int f(x-y,t)\,dy \tag{4.19}$$

Notre objectif était donc de passer en revue les algorithmes de traitement d'images utilisant les automates cellulaires et pouvant être appliqués aux images de documents pour chercher une méthode pouvant bénéficier des nouveautés introduites par *Smooth Life*. Malheureusement devant l'absence de candidats satisfaisants à ces critères, nous avons finalement décidé d'abandonner cette piste et de mettre au point une méthode originale.

Cette méthode utilise l'astuce de calcul du taux de remplissage normalisée dans un cercle proposée par l'auteur de *Smooth Life*. Nous l'appliquons pour chaque pixel de l'image en faisant varier la taille du cercle utilisé. Nous construisons ainsi un cube de données comme illustré par la figure 4.3. Pour chaque pixel nous disposons alors d'un vecteur d'information. L'hypothèse sur laquelle nous nous reposons ensuite est que les différentes zones d'une image de document (texte, dégradations, fond, etc.) présenteront des réponses différentes et discriminantes. Un aperçu est montré à l'aide de la figure 4.4 Un algorithme très simple en six étapes est alors proposé :

1. Calcul du cube, pour des cercles de diamètres allant de n à m,
2. calcul des vecteurs différentiels,
3. partitionnement des vecteurs par l'algorithme des K-moyennes en cinq classes différentes
4. extraction de la classe de texte et d'une classe de diffusion ; la classe de texte est celle dont la moyenne des valeurs de pixels est la plus faible. La classe de diffusion est celle qui est le plus en contact avec la précédente.
5. diffusion par inondation de la classe de texte dans la classe de diffusion,
6. seuillage des valeurs et binarisation.

La figure 4.5 illustre le fonctionnement de l'algorithme dans sa répartition en classes d'une image.

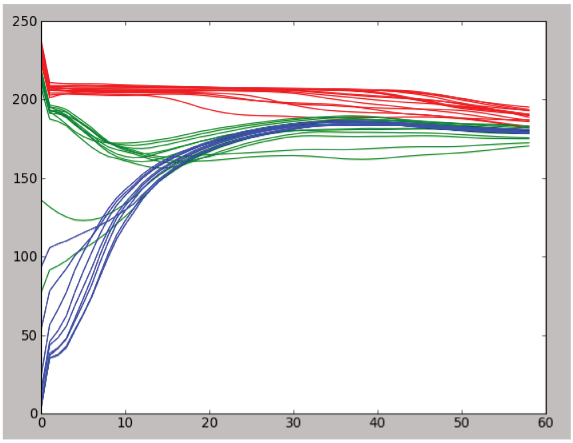


FIGURE 4.4 – Exemple de vecteurs caractéristiques selon l'emplacement du pixel étudié : les réponses des pixels appartenant au fond de l'image sont en rouge, celles des contours d'une zone de texte en vert et celles correspondant à des pixels de texte en bleu.

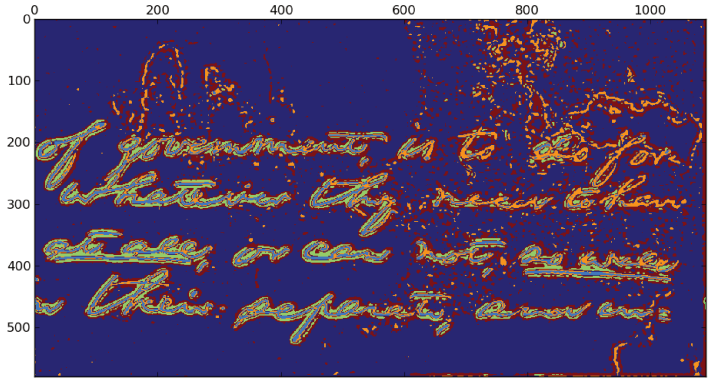


FIGURE 4.5 – Segmentation cinq classes par k-moyennes à partir des vecteurs caractéristiques du prototype proposé d'un document fortement dégradé. Le bleu clair correspond à la classe de texte à diffuser dans l'orange. Le résultat final est illustré avec la figure 4.6.

Nous évaluons cet algorithme en l'appliquant sur des images de documents fortement dégradées issues du concours international DIBCO. La figure 4.6 montre quelques résultats qui résument ses capacités.

C'est par ce travail que nous avons abordé ces travaux de thèse afin de rentrer dans le sujet. Aussi, à l'heure où nous y apportons la conclusion, il nous est difficile de ne pas remarquer les « nombreuses erreurs de jeunesse » qui y sont présentes. Les résultats ne sont globalement guère mieux que des approches au seuillage global mais donnent parfois des surprises positives comme pour la figure 4.5. L'absence totale de réflexion quant à la notion d'échelle est particulièrement regrettable ; après notre état de l'art sur les détecteurs de points d'intérêt il nous parait indispensable de réaliser une analyse multi-échelles. Encore plus préjudiciable, nous ne proposons aucune étape de « nettoyage » qui sont pourtant fréquentes dans la littérature ; des petits points peuvent engendrer lors de la diffusion des faux positifs importants. Et pour terminer, l'utilisation du calcul du taux de remplissage comme utilisée par *smooth life* est superflue car l'approche est très proche d'une utilisation de différences de gaussiennes qui peuvent êtres obtenues très rapidement en ayant recours à des images intégrales.

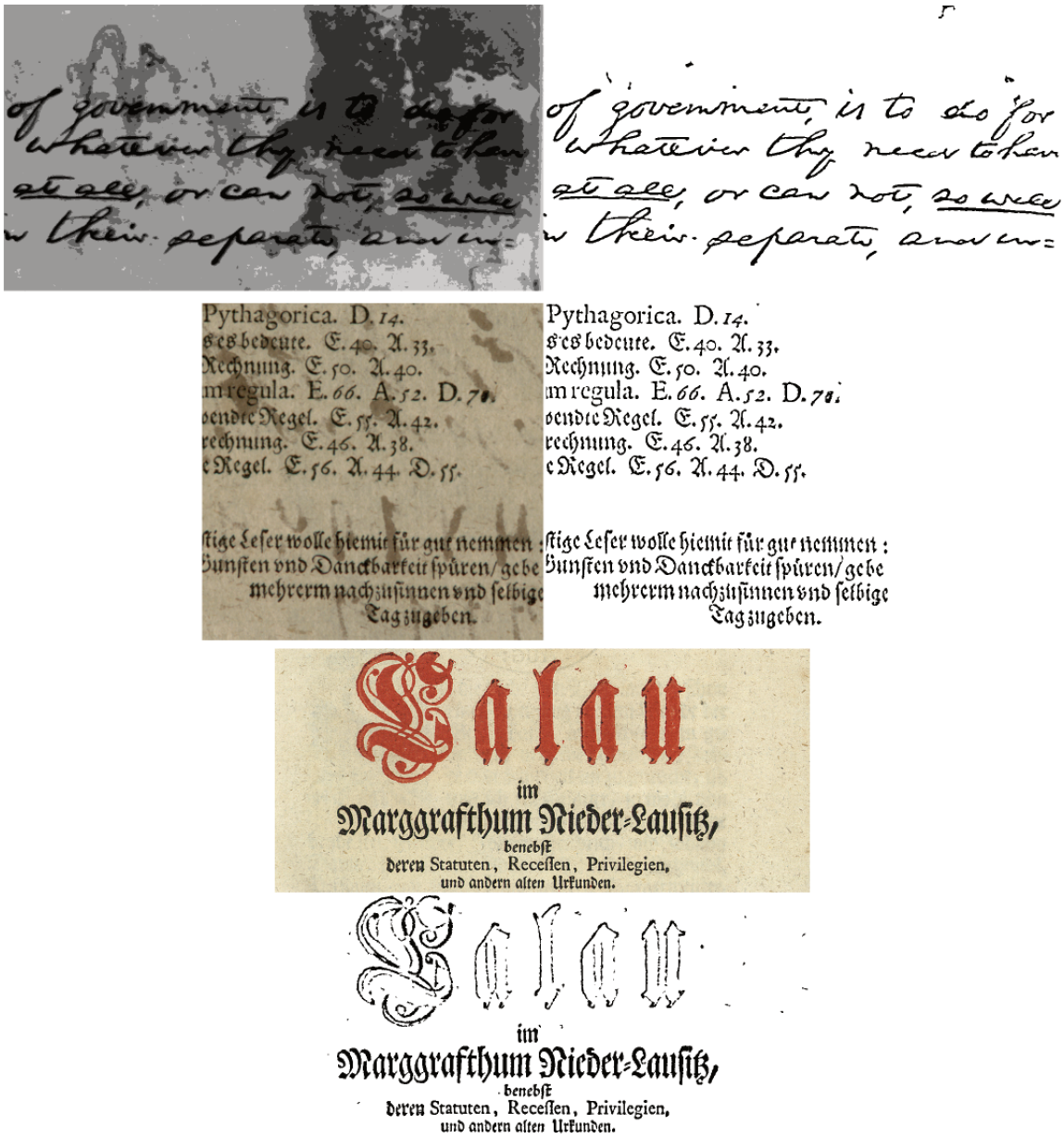


FIGURE 4.6 – Quelques résultats mitigés de ce prototype de binarisation sur des images issues de DIBCO. Des problèmes d'échelles apparaissent (en bas) mais aussi, plus grave, des résidus comme nous en observons habituellement sur des méthodes de seuillage global.

4.3 Extraction par analyse de la structure

4.3.1 Introduction

Les détecteurs classiques de points d'intérêt ne sont peut-être pas bien adaptés aux images de documents mais ils sont conçus pour posséder une très forte répétabilité ainsi qu'une précision au pixel près. Ces caractéristiques sont particulièrement appréciables et il serait regrettable de s'en passer. C'est pourquoi, plutôt que de s'aventurer dans la conception d'un tout nouveau détecteur nous allons tâcher de guider les approches classiques en réalisant une détection en amont de zones d'intérêts sur lesquelles nous appliquerons les analyses de saillances habituelles.

La structure d'un document texte possède une information riche. Celui-ci est constitué de titres, sous-titres, paragraphes, retours à la ligne... etc. qui fluidifient la lecture pour l'œil humain qui a appris culturellement à analyser rapidement cette structure, indiquant comment est constituée l'information. Tout un chacun peut constater qu'en balayant rapidement un document du regard, ce dernier est sensiblement attiré par ces éléments de structure. Aussi nous faisons l'hypothèse que ce sont des zones riches d'informations visuelles pour l'extraction de descripteurs caractéristiques. Nous avons vu la profusion de points d'intérêt retournés dans une image de documents : nous allons tenter de réduire cet ensemble en le restreignant à la structure du document.

4.3.2 Méthode

Chronologiquement dans l'élaboration de notre travail, cette idée est apparue avant la réalisation de notre algorithme CORE mais nous avons opté pour le developpement de ce dernier en premier. Nous pouvons a posteriori avoir un élément de réponse quant à la pertinence de notre idée en inspectant visuellement les résultats du filtrage par CORE avec la figure 4.7. Nous en observons une certaine confirmation ; les titres et sous-titres semblement contenir plus de points pertinents mais aussi les bordures de paragraphes.

Nous proposons la méthode suivante -très simple- pour extraire un masque de ces zones pertinentes. Tout d'abord, l'image capturée est traitée selon une étape de binarisation : l'objectif étant de marquer les pixels correspondant à des éléments imprimés en noir et tout ceux qui correspondent au papier en blanc. Nous avons vu précédemment que le domaine de la binarisation est vaste et que de nombreux algorithmes existent. Pour des images de haute qualité comme celles produites lors de l'acquisition sous conditions totalement maitrisées comme avec un scanner, nous pourrions utiliser des méthodes de seuillage global tel que Otsu. Mais nous nous intéressons dans notre étude à la capture nomade, il est donc nécessaire de prendre en considération des variations d'intensité lumineuse, du bruit, etc. Il est préférable de recourir au moins à des méthodes de seuillage locaux comme celle de Sauvola. Notons aussi que pour des images dont les documents sont particulièrement abimés, nous pourions utiliser des approches plus complexes comme l'algorithme FAIR qui produit des résultats remarquables dans ces cas tout en conservant des temps de calculs raisonnables.

Une fois l'image binaire obtenue, nous y appliquons des outils issus de la morphologie mathématique pour construire notre masque. Une première étape de dilatation est appliquée à l'aide de l'équation :

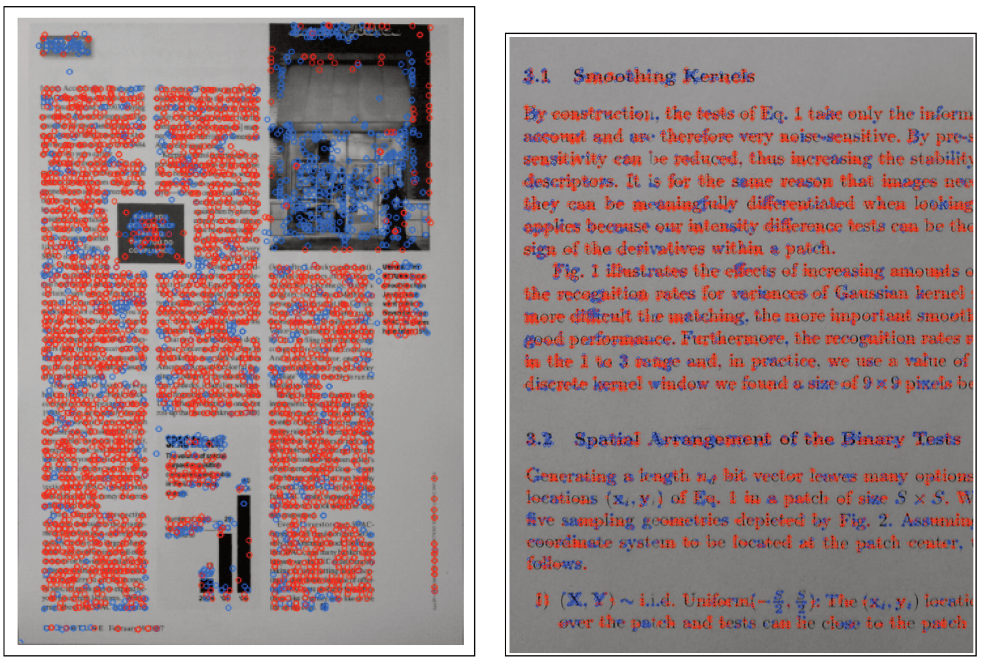


FIGURE 4.7 – Application du filtrage CORE sur des points d'intérêt et descripteurs SIFT. A gauche, extrait de la base de données SmartDOC, à droite, image personnelle capturée à partir d'un *smartphone*. Les points conservés sont en bleu, ceux mis de côté en rouge.

$$\mathrm{D}il_{\mathrm{K}}(\mathrm{I}) = \cup\{\mathrm{K}_p | p \in \mathrm{I}\} \tag{4.20}$$

où I est l'image entrée, p un pixel à l'intérieur de celle-ci et K le noyau matrice 3×3 suivant :

$$\mathrm{K} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Cela a pour effet de connecter les mots présents dans le texte comme des composantes entièrement connectées, fusionnant les petits éléments de ponctuation tels que les accents ou les apostrophes. Ensuite, une deuxième étape de dilatation est appliquée avec un noyau différent :

$$\mathrm{K} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Nous relions le nombre d'itérations de ces opérations à la dimension moyenne d'une lettre. Celle-ci est trivialement obtenue par une analyse en amont des composantes connexes juste après la phase de binarisation.

En procédant ainsi, nous fusionnons les mots alignés et formons des lignes. Nous récupérons ces nouvelles composantes connexes et procédons à de rapides analyses basées sur leurs statistiques : celles dont l'épaisseur est très faible sont très certainement des points résultant de bruits, des artefacts de la phase de binarisation qui n'ont pas été supprimés par la première dilatation. Nous les supprimons. Nous séparons celles qui restent en deux catégories, selon leur longueur. La majorité des composantes connexes ont approximativement la même taille, légèrement plus petites que la largeur de la page : ce sont

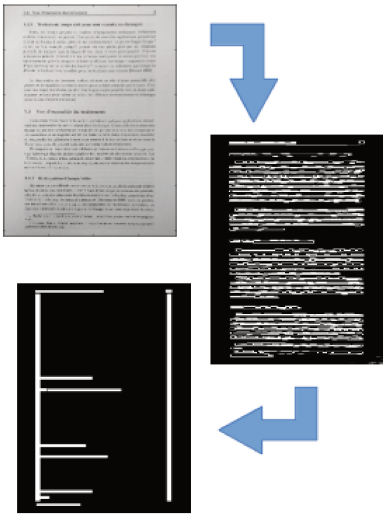


FIGURE 4.8 – Processus de construction d'un masque de localisation de zones d'intérêt pour l'extraction de points clefs par binarisation puis application de méthodes de morphologie mathématique avant analyse des composantes connexes résultantes pour la constitution d'un masque des emplacements de débuts et fins de lignes et titres / sous-titres.

les lignes constituant le corps du texte. La deuxième catégories rassemble des lignes plus petites que nous considérons comme des titres, des sous-titres, voire des fins de paragraphes.

A partir de la première classe, celle des lignes pleines, nous calculons la position moyenne du début et de la fin des lignes. Avec cette information, nous pouvons débuter un traçage de masque avec deux colonnes qui entourent les bords de lignes. Ensuite, en suivant nos hypothèses et les constatations visuelles de l'algorithme CORE, nous utilisons la totalité des emplacements de la seconde classe pour compléter ce masque. A partir de cette étape, nous possédons un masque qui nous renseigne sur l'endroit où nous allons appliquer les algorithmes d'extraction de points d'intérêt dans l'image sans avoir à analyser la saillance de chaque pixel. Tout ce processus est illustré par la figure 4.8.

Avant de continuer, il nous parait nécessaire de souligner les limites de cette proposition. Nous ne nous sommes pas intéressés à tout ce qui peut être en rapport avec l'estimation de l'orientation du document. Notre approche nécessite donc d'avoir une capture du document relativement droite (une légère orientation n'est pas problématique). De plus nous ne proposons pas de segmentation du document sur un fond, l'image ne doit donc contenir que le document et limiter la capture d'autres éléments. Enfin, le document doit être photographié dans son entier, nous ne traitons pas le cas où celui-ci est morcelé. Nous attirons l'attention du lecteur sur le fait que cette proposition est plus une preuve de concept qui est apparue vers la fin de nos travaux ; si, comme nous allons le voir, les résultats sont encourageants il sera alors pertinent de développer l'idée plus en profondeur pour prendre en compte les remarques que nous venons de faire.

Mise en correspondance précoce

Néanmoins, ces faiblesses ne nous empêchent pas d'explorer plus en profondeur l'utilisation de la structure du document. Nous proposons de réaliser la mise en correspondance des points d'intérêt en deux étapes. Dès l'obtention du masque d'extraction dans

chaque image nous pouvons appareiller les composantes connexes entre elles grâce à leurs caractéristiques que sont leurs tailles et emplacements dans l'image. Nous construisons de petits vecteurs caractéristiques pour chaque composante, constitués des coordonnées en x et en y de leurs emplacements ainsi que de leurs dimensions. L'appariement se fait par force brute en utilisant toutefois le ratio de Lowe ($d = 0.4$).

Cela nous permet de construire des sous-ensembles de points d'intérêt et d'éviter les dispersions, renforçant une certaine cohérence spatiale. Les points d'intérêts sont par la suite appareillés uniquement avec ceux du sous-ensemble correspondant dans la deuxième image. Si les composantes connexes ont été correctement mises en correspondances, cela devrait nous apporter une augmentation non négligeable du *inlier ratio*. Nous testons cette idée d'appariement précoce de deux façons : avec l'ensemble des composantes connexes et avec seulement les petites (correspondant aux titres, sous-titres, etc).

4.3.3 Evaluations et résultats

Pour évaluer cette proposition, nous utilisons une fois de plus la métrique du *inlier ratio* comme employée précédemment pour évaluer notre algorithme CORE. Après l'extraction des points d'intérêt dans un couple d'images d'une même scène, nous calculons un appariement de ces points par méthode force-brute selon une distance d'Euclide. A partir de ces correspondances, nous appliquons l'algorithme RANSAC afin de déterminer la transformation sous-jacente entre ces deux images et nous évaluons le ratio des correspondances cohérentes avec ce modèle sur le total.

Cette évaluation se base sur trois descripteurs populaires, à savoir : SIFT, BRISK et ORB. Nous comparons cette proposition à notre algorithme CORE et une approche de référence qui est celle classique sans aucun filtrage. En ce qui concerne ORB, puisque sa méthode de détection de points d'intérêt est légèrement différente des extracteurs classiques (en imposant un nombre fixe de points à retourner, triés par une mesure de Harris), nous la remplaçons par le détecteur SURF qui est similaire à celui de SIFT. Cela nous donne des comparaisons plus objectives. Encore une fois, nous appliquons le classique test du ratio de Lowe afin d'éviter d'avoir à prendre en compte des ensembles disjoints pouvant polluer cette évaluation ($d = 0.7$).

En ce qui concerne le choix des données à utiliser, malheureusement, admettant les faiblesses de notre méthode qui est assez simple dans son approche d'analyse de la structure du document, nous sommes obligés de constater que nous ne pouvons pas en l'état l'appliquer sur SmartDOC. Nous utilisons à nouveau notre ensemble personnel d'images capturées par *smartphone*. Chaque couple d'images contient un document différent, issu de publications scientifiques sans illustrations.

Les résultats sont présentés par le biais de la figure 4.9. Nous observons que notre contribution augmente dans les trois cas étudiés le *inlier ratio* moyen. Nous ne pouvons pas affirmer la même chose en ce qui concerne l'algorithme CORE qui nous donne des résultats sensiblement similaires à l'approche de référence pour SIFT et BRISK. Toutefois, nous insistons sur le fait que ce n'est pas un mauvais résultat puisque le nombre total de points a bien été drastiquement réduit, ce qui implique une convergence plus rapide de RANSAC comme illustrée par le tableau 4.1 : comme attendu, la phase de détection

Feature	Extraction		*speed up*	RANSAC		*speed up*
SIFT	3.62	2.48	*1.46*	18.89	0.18	*102.0*
BRISK	0.67	0.32	*2.10*	11.20	0.12	*90.3*
ORB	0.47	0.36	*1.32*	4.30	0.03	*135.1*

TABLEAU 4.1 – Comparaison des temps de traitements en secondes de l'extraction de points d'intérêt et de l'estimation RANSAC selon le descripteur et l'approche utilisés. Pour chaque colonne, la gauche et la droite sont respectivement les valeurs références et la méthode proposée.

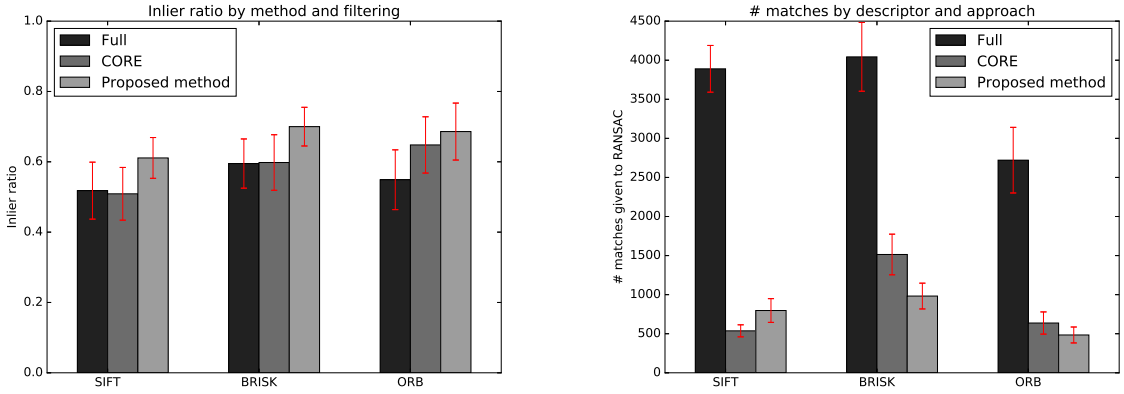


FIGURE 4.9 – A gauche, *inlier ratio* moyen de l'estimation RANSAC par descripteur et approche employée. A droite, nombre total de correspondances données à l'algorithme RANSAC.

est certes plus rapide mais le plus grand bénéfice vient de l'estimation par RANSAC qui bénéficie d'une réduction de presque 19 secondes à seulement 0.2 secondes.

En ce qui concerne notre idée de mise en correspondance précoce, les résultats sont présentés avec la figure 4.10. Nous observons systématiquement un gain du *inlier ratio* pour tous les descripteurs, nous apportant des résultats encore meilleurs que précédemment avec ORB bénéficiant le plus de cette proposition (55% pour la référence à 80% pour notre contribution). De surcroit, nous remarquons que les ensembles de correspondances sont plus importants, ce qui impliquerait que moins de couples ont été écartés par le test de Lowe. Ceci confirme l'aspect de réduction de la confusion recherché. De la même façon, le nombre de correspondances par pre-appariement des petites composantes connexes seulement est à peine plus faible que l'approche précédente en dépit d'une diminution importante du nombre brut de points d'intérêt.

4.3.4 Conclusions

Nous nous sommes intéressés à l'idée de guider les algorithmes classiques en vision d'extraction de points d'intérêt sur des images de documents par analyse de la structure. Nous avons comparé notre proposition à notre algorithme CORE pour la réduction de la confusion. Les résultats ont montré des améliorations dans la qualité des ensembles de correspondances et une convergence plus rapide de RANSAC par rapport aux valeurs références. De surcroit, nous avons constaté que des idées très simples comme la mise en correspondance de sous-ensembles de points d'intérêt pouvaient contribuer à améliorer encore plus les résultats.
Tout cela semble confirmer notre hypothèse première et nous encourage à poursuivre ces

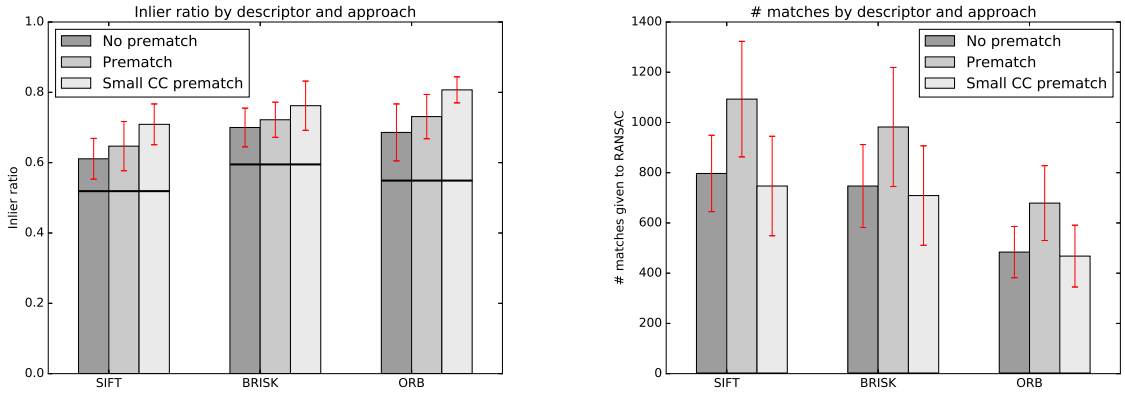


FIGURE 4.10 – A gauche, *inlier ratio* moyen des différentes façons de réaliser notre proposition. Les barres horizontales correspondent aux valeurs références pour chaque descripteur.
A droite, nombre total de correspondances données à l'algorithme RANSAC.

développements plus loin avec des méthodes plus robustes, comme évoqué précédemment.

4.4 Conclusions

Une image de document présente des spécificités très particulières. Nous avons vu auparavant l'inadéquation des détecteurs classiques aux problèmes dédiés à la communauté document. Dans ce chapitre nous nous sommes intéressés aux travaux spécialement conçus pour cette problématique et nous avons vu comment les contributions tiraient parti astucieusement des caractéristiques du document telle que l'utilisation des centroïds de mots pour la détection de points d'intérêt. Dans notre objectif de réaliser le pont entre les algorithmes classiques de la vision par ordinateur et les problèmes liés au document, nous avons proposé de guider l'extraction des points d'intérêt en s'appuyant sur la structure du document. Les résultats, bien qu'ayant une porté limitée, confirment l'intérêt de réaliser un tel pont qui devra être étudié plus en profondeur à l'avenir.

Chapitre 5

Conclusions et perspectives

5.1 Bilan

Nous sommes arrivés au terme de cette étude et le moment est venu d'établir un bilan de nos travaux. Nous nous sommes efforcés de concilier les algorithmes de détection de points d'intérêt et d'extraction de descripteurs associés avec les images de documents qui n'ont pas été prises en compte lors de l'élaboration de ceux-ci. Notre principale contribution est l'algorithme de réduction de la confusion (CORE) qui filtre les descripteurs locaux dont le risque de confusion est supérieur à un seuil fixé par l'utilisateur. Les mérites de cette approche sont multiples. Nous avons été guidés dans son élaboration par un souci de simplicité et de généricité ; Cette méthode présente ainsi l'intéret de pouvoir être mise en oeuvre sur une image isolée ce qui permet d'envisager de l'utiliser pour des problèmes d'indexation ou de reconnaissance (qui utilisent également les decripteurs) sans être limité à la mise en correspondance.

D'autre part, bien qu'initialement développé dans le contexte des images de documents, l'algorithme CORE trouve légitimement des applications à toutes les situations génératrices de confusion (scènes urbaines, etc.). De plus, le calcul du seuil de probabilité se fait à partir d'une valeur p fixée par l'utilisateur : celle-ci est une probabilité et par conséquent le nombre de points filtrés dépend uniquement du taux de confusion présent dans l'ensemble de points étudié ce qui permet une facilité d'utilisation très appréciable, contrairement aux approches telles que le détecteur ORB qui demande explicitement le nombre de points à retourner dans l'image.

Les résultats ont montré les bénéfices de cette méthode, que ce soit dans le ratio des correspondances cohérentes avec le modèle dégagé ou dans le temps de calcul de l'estimation d'une homographie, considérablement raccourcie puisque la convergence se fait plus rapidement. Nous avons par ailleurs utilisé le jeu de données SmartDoc pour avoir la garantie d'une justification statistique solide quant aux mérites de notre contribution. Comme alternative, toutefois moins souple, nous avons tenté de bénéficier de la structure du document pour guider l'extraction des points d'intérêt. Cette approche est confirmée visuellement par les résultats que nous obtenons avec CORE et bien que d'une portée limitée, considérant le jeu de données utilisé et la simplicité de la méthode, les résultats sont très satisfaisants, souvent supérieurs à l'application de CORE. Mais bien évidemment nous ne nous arrêterons pas là.

5.2 Perspectives

Un travail de recherche est forcément incomplet et nous irons même jusqu'à dire nécessairement, en référence au théroème d'incomplétude de Kurt Gödel qui peut se traduire par « il existera toujours des questions sans réponses ». Nous présentons ici quelques pistes de poursuites de nos travaux.

5.2.1 Amélioration de l'algorithme CORE

Une limitation sévère de l'algoritme CORE actuel est de mettre sur un pied d'égalité toutes les dimensions des descripteurs. La variance σ et la probabilité de basculement d'un bit μ sont identiques pour toutes les dimensions d'un descripteur, notre analyse étant une moyenne. Or Balntas *et al.* ont prouvé avec BOLD en 2015 que les dimensions ne sont pas toutes égales dans leur capacité à garder l'information de manière stable. Si nous pouvions réaliser une étude *offline* poussée pour déterminer les valeurs σ et μ pour chaque dimension d'un descripteur, le filtrage devrait être bien plus fidèle par rapport à la réalité physique de la confusion. Ces développements nécessiteront toutefois une réécriture des équations du calcul du critère et de l'estimation du seuil.

5.2.2 Optimisations du temps de calcul

Dans le chapitre 3, nous avons présenté une optimisation de l'algorithme pour architecture GPU. Nous présentons ici une idée d'amélioration générique qui sera utile pour les processeurs classiques et que nous explorerons davantage dans le futur. Elle repose sur des approximations, la figure 5.1 en est une illustration. Dans l'espace des descripteurs, il est probable -surtout si nous avons à traiter un problème de confusion- que des vecteurs soient très proches les uns des autres. L'idée, selon un processus totalement séquentiel, serait de vérifier au moment du calcul d'un critère si un autre vecteur proche n'a pas déjà été évalué. Si c'est bien le cas, le calcul n'est pas réalisé et la valeur du critère du point voisin est utilisée.

La pertinence de cette proposition repose sur l'existence d'une structure de données d'indexations de vecteurs et de recherches de plus proches voisins qui soit plus rapide dans son utilisation que la réalisation du calcul du critère. De bons candidats sont à rechercher dans la librairie FLANN spécialisée dans la recherche de voisins par approximations. Bien évidemment, des études seront nécessaires pour trouver le juste milieu du seuil de proximité : s'il est trop fin, le gain de temps de calcul sera négligeable (voire négatif puisqu'il faut prendre en compte la phase d'indéxation) et si à l'inverse il est trop grossier, les critères retournés ne seront pas fidèles à la réalité et le filtrage en serait fortement pénalisé.

5.2.3 Analyse de l'entropie

Nous posons ici des réflexions exploratoires quant à l'élaboration d'un critère d'efficacité du filtrage basé sur l'évaluation de la quantité d'information constituée par l'ensemble des descripteurs calculés sur l'image. Cette mesure pourrait aussi être utilisé pour évaluer le degré de confusion présent dans l'ensemble de descripteurs avant d'appliquer l'algorithme CORE.

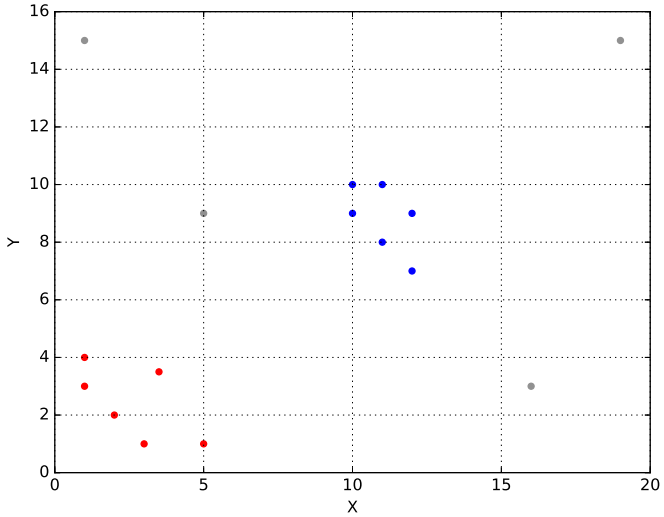


FIGURE 5.1 – Répartition de vecteurs pour un descripteur fictif à deux dimensions, X et Y. La présence de groupes (en couleurs) permet d'attribuer un même critère sans avoir à réaliser un calcul.

Afin de caractériser cette information, nous utiliserons classiquement une mesure d'entropie et plus particulièrement l'entropie de Rényi qui généralise les définitions habituellement rencontrées (Hartley, Shannon, de collision, etc.). Pour une variable aléatoire continue $\mathbf{u}$, la définition générale de l'entropie de Rényi est donnée par :

$$H_\alpha(\mathbf{u}) = \frac{1}{1-\alpha} \log\left(\int P_{\mathbf{u}}(\mathbf{v})^\alpha d\mathbf{v}\right) \tag{5.1}$$

et

$$H_\alpha(\mathbf{u}) = \frac{1}{1-\alpha} \log\left(\sum_{i=1}^{n} p(\mathbf{u_i})^\alpha\right) \tag{5.2}$$

pour une variable aléatoire discrète, qui correspond à l'utilisation sur des vecteurs binaires. Dans notre problématique, nous nous intéressons surtout au cas particulier où $\alpha = 2$ qui est celui de l'entropie de collision, tout à fait pertinent pour analyser la quantité d'informations d'un ensemble de descripteurs dans l'étude de la confusion.

Dans le cas des descripteurs flottants, l'expression de l'entropie de collision associée à l'ensemble des descripteurs calculés sur une image se déduit simplement de la définition initiale :

$$\begin{aligned}
H_2(\mathbf{u}) &= -\log\left(\int \left(\frac{1}{(\mathbf{N})\left(\sigma\sqrt{2\pi}\right)^D} \sum_{\mathbf{i}=1}^{N} \exp\left(-\frac{d_E(\mathbf{u},\mathbf{u_i})^2}{2\sigma^2}\right)\right)^2 d\mathbf{u}\right) && (5.3)\\
&= -\log\left(\int \left(\frac{1}{\mathbf{N}} \sum_{\mathbf{i}=1}^{N} G_\sigma\left(d_E(\mathbf{u},\mathbf{u_i})\right)\right)^2 d\mathbf{u}\right) && (5.4)\\
&= -\log\left(\frac{1}{\mathbf{N}^2} \sum_{\mathbf{i}=1}^{N}\sum_{\mathbf{j}=1}^{N} \int G_\sigma\left(d_E(\mathbf{u},\mathbf{u_i})\right) G_\sigma\left(d_E(\mathbf{u},\mathbf{u_j})\right) d\mathbf{u}\right) && (5.5)\\
&= -\log\left(\frac{1}{\mathbf{N}^2} \sum_{\mathbf{i}=1}^{N}\sum_{\mathbf{j}=1}^{N} G_{\sigma\sqrt{2}}\left(d_E(\mathbf{u_i},\mathbf{u_j})\right)\right) && (5.6)
\end{aligned}$$

Dans le cas binaire, l'application de la définition de l'entropie de Rény à la distribution de probabilité de l'ensemble des descripteurs s'écrit :

$$\mathrm{H}_2(\mathbf{u}) = -\log\left(\sum_{\mathbf{u}}\left(\frac{1}{\mathbf{N}}\sum_{\mathbf{i}=1}^{\mathrm{N}}\mu^{\mathrm{d_H}(\mathbf{u},\mathbf{u_i})}(1-\mu)^{\mathrm{D}-\mathrm{d_H}(\mathbf{u},\mathbf{u_i})}\right)^2\right) \tag{5.7}$$

$$= -\log\left(\frac{1}{\mathbf{N}^2}\sum_{\mathbf{i}=1}^{\mathrm{N}}\sum_{\mathbf{j}=1}^{\mathrm{N}}\sum_{\mathbf{u}}\mu^{\mathrm{d_H}(\mathbf{u},\mathbf{u_i})+\mathrm{d_H}(\mathbf{u},\mathbf{u_j})}(1-\mu)^{2\mathrm{D}-\mathrm{d_H}(\mathbf{u},\mathbf{u_i})-\mathrm{d_H}(\mathbf{u},\mathbf{u_j})}\right) \tag{5.8}$$

En raison de la sommation réalisée sur l'ensemble de l'espace des configurations, cette dernière équation n'est cependant pas calculable pratiquement.

Une alternative consiste à dénombrer le nombre de configurations possibles de $\mathbf{u}$ qui correspondent à une distance K donnée, la sommation se trouve ainsi réalisée sur le nombre de distances possibles.

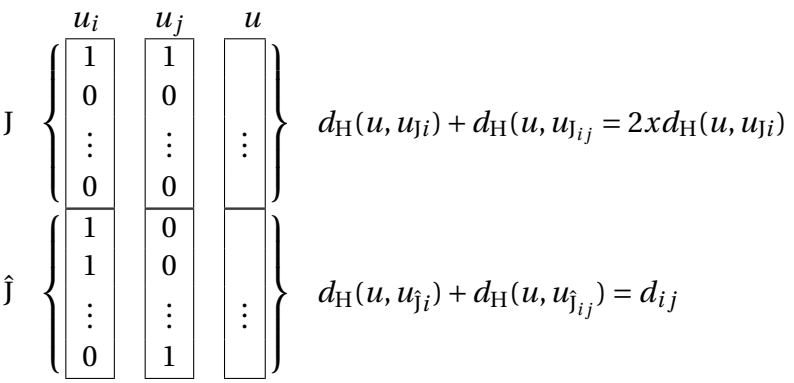


FIGURE 5.2 – Puisque nous traitons des distances de Hamming, nous pouvons réorganiser les bits entre u_i et u_j comme nous le souhaitons.

Définissons maintenant le partitionnement suivant de l'ensemble des composantes du descripteur binaire. Soit $\mathrm{J} = \{\mathrm{J}_1, \ldots, \mathrm{J}_{d_{ij}}\}$ le plus grand sous-ensemble de $\{1, \ldots, \mathrm{D}\}$ tel que $\mathrm{d_H}(\mathbf{u_{iJ}}, \mathbf{u_{jJ}}) = 0$ (autrement dit, l'ensemble des bits identiques entre $\mathbf{u_i}$ et $\mathbf{u_j}$). Soit aussi $\bar{\mathrm{J}}$ le complémentaire de J dans $\{1, \ldots, \mathrm{D}\}$. Le nombre de bits de $\bar{\mathrm{J}}$ correspond à la distance d_{ij}. Ainsi, indépendamment de la configuration de u côté $\bar{\mathrm{J}}$, la somme des deux distances $\mathrm{D}_h(u, u_i)$ et $\mathrm{D}_h(u, u_j)$ vaut toujours d_{ij}.
Nous obtenons finalement l'expression suivante :

$$\mathrm{H}_2(\mathbf{u}) = -\log\left(\frac{1}{\mathbf{N}^2}\sum_{\mathbf{i}=1}^{\mathrm{N}}\sum_{\mathbf{j}=1}^{\mathrm{N}}\sum_{k=0}^{\mathrm{D}-d_{ij}} 2^{d_{ij}}\binom{\mathrm{D}-d_{ij}}{k}\mu^{2k+d_{ij}}(1-\mu)^{2\mathrm{D}-2k-d_{ij}}\right) \tag{5.9}$$

Nous disposons ainsi d'un outil de mesure de l'entropie de collision d'une distribution de vecteurs caractéristiques. Ce travail nous sera utile soit pour analyser l'efficacité du filtrage, soit pour paramétrer les réglages de l'algorithme.

Bibliographie

[Agrawal and Doermann, 2009] Agrawal, M. and Doermann, D. (2009). Voronoi++ : A dynamic page segmentation approach based on voronoi and docstrum features. In *2009 10th International Conference on Document Analysis and Recognition*, pages 1011–1015. 88

[Agrawal et al., 2008] Agrawal, M., Konolige, K., and Blas, M. R. (2008). *CenSurE : Center Surround Extremas for Realtime Feature Detection and Matching*, pages 102–115. Springer Berlin Heidelberg, Berlin, Heidelberg. 19

[Alahi et al., 2012] Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak : Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517. iii, 20, 36, 37

[Baird, 1992] Baird, H. S. (1992). Background structure in document images. In *In Advances in Structural and Syntactic Pattern Recognition*, pages 17–34. World Scientific. 90

[Balntas et al., 2015] Balntas, V., Tang, L., and Mikolajczyk, K. (2015). Bold - binary online learned descriptor for efficient image matching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2367–2375. iii, 40, 41

[Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3) :346 – 359. Similarity Matching in Computer Vision and Multimedia. 19, 26

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4) :509–522. ii, 24, 25

[Block et al., 2007] Block, M., Ortegón, M. R., Seibert, A., Kretzschmar, J., and Rojas, R. (2007). Sitt - a simple robust scaleinvariant text feature detector for document mosaicing. 84

[Block and Rojas, 2009] Block, M. and Rojas, R. (2009). Local contrast segmentation to binarize images. In *2009 Third International Conference on Digital Society*, pages 294–299. 93

[Breuel, 2002] Breuel, T. M. (2002). *Two Geometric Algorithms for Layout Analysis*, pages 188–199. Springer Berlin Heidelberg, Berlin, Heidelberg. 90

[Burie et al., 2015] Burie, J., Chazalon, J., Coustaty, M., Eskenazi, S., Luqman, M., Mehri, M., Nayef, N., Ogier, J., Prum, S., and nol, M. R. (2015). ICDAR2015 competition on smartphone document capture and OCR (smartdoc). In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 1161–1165. 72

[C. Strecha and Fua, 2012] C. Strecha, A. M. Bronstein, M. M. B. and Fua, P. (2012). LDA-Hash : Improved Matching with Smaller Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1). iii, 45

[Calonder et al., 2010] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). *BRIEF : Binary Robust Independent Elementary Features*, pages 778–792. Springer Berlin Heidelberg, Berlin, Heidelberg. iii, 30, 31

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6) :679–698. 93

[Cao et al., 2000] Cao, R., Tan, C. L., Wang, Q., and Shen, P. (2000). Segmentation and analysis of double-sided handwritten archival documents. 93

[Cesarini et al., 2001] Cesarini, F., Lastri, M., Marinai, S., and Soda, G. (2001). Encoding of modified x-y trees for document classification. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pages 1131–1136. 90

[Chazalon et al., 2015] Chazalon, J., Rusiñol, M., and Ogier, J. M. (2015). Improving document matching performance by local descriptor filtering. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1216–1220. iv, 52, 53, 73

[Chen et al., 2008] Chen, Q., Sun, Q.-s., Ann Heng, P., and Xia, D.-s. (2008). A double-threshold image binarization method based on edge detector. *Pattern Recognition*, 41(4) :1254–1267. 93

[Corbelli et al., 2016] Corbelli, A., Baraldi, L., Grana, C., and Cucchiara, R. (2016). Historical document digitization through layout analysis and deep content classification. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 4077–4082. 90

[Crow, 1984] Crow, F. C. (1984). Summed-area tables for texture mapping. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pages 207–212, New York, NY, USA. 19

[Dang et al., 2015] Dang, Q. B., Luqman, M. M., Coustaty, M., Tran, C. D., and Ogier, J. M. (2015). Srif : Scale and rotation invariant features for camera-based document image retrieval. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 601–605. 85

[Dang et al., 2016] Dang, Q. B., Rusiñol, M., Coustaty, M., Luqman, M. M., Tran, C. D., and Ogier, J. M. (2016). Delaunay triangulation-based features for camera-based document image retrieval system. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 1–6. 85

[Delaunay, 1934] Delaunay, B. (1934). Sur la sphère vide. a la mémoire de georges voronoï. *Bulletin de l'Académie des Sciences de l'URSS*, (6) :793–800. 85

[Ebrahimi and Mayol-Cuevas, 2009] Ebrahimi, M. and Mayol-Cuevas, W. W. (2009). Susure : Speeded up surround extrema feature detector and descriptor for realtime applications. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–14. 19

[Everingham et al.,] Everingham, M., Zisserman, A., Williams, C. K. I., and Van Gool, L. The pascal visual object classes challenge 2006 (voc2006) results. http ://www.pascal-network.org/challenges/VOC/voc2006/results.pdf. 33

[Fabrizio et al., 2009] Fabrizio, J., Marcotegui, B., and Cord, M. (2009). Text segmentation in natural scenes using toggle-mapping. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 2373–2376. 94

[Fellows and Koblitz, 2000] Fellows, M. R. and Koblitz, N. (2000). Combinatorially based cryptography for children (and adults). 4

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6) :381–395. 52, 55

[Frome et al., 2004] Frome, A., Huber, D., Kolluri, R., Bülow, T., and Malik, J. (2004). *Recognizing Objects in Range Data Using Regional Point Descriptors*, pages 224–237. Springer Berlin Heidelberg, Berlin, Heidelberg. 25

[Gabor, 1946] Gabor, D. (1946). page 429–457. 38

[Gao et al., 2014] Gao, H., Rusiñol, M., Karatzas, D., and Lladós, J. (2014). Embedding document structure to bag-of-words through pair-wise stable key-regions. In *2014 22nd International Conference on Pattern Recognition*, pages 2903–2908. 86

[Gao et al., 2013] Gao, H., Rusiñol, M., Karatzas, D., Lladós, J., Sato, T., Iwamura, M., and Kise, K. (2013). Key-region detection for document images – application to administrative document retrieval. In *2013 12th International Conference on Document Analysis and Recognition*, pages 230–234. 86

[Garey, 1972] Garey, M. R. (1972). Optimal binary identification procedures. *SIAM Journal on Applied Mathematics*, 23(2) :173–186. 17

[Gatos et al., 2006] Gatos, B., Pratikakis, I., and Perantonis, S. J. (2006). Adaptive degraded document image binarization. *Pattern Recogn.*, 39(3) :317–327. 94

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151. 11

[Hinds et al., 1990] Hinds, S. C., Fisher, J. L., and D'Amato, D. P. (1990). A document skew detection method using run-length encoding and the hough transform. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, volume i, pages 464–468 vol.1. 88

[Hyafil and Rivest, 1976] Hyafil, L. and Rivest, R. L. (1976). Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1) :15 – 17. 16

[Jaccard, 1901] Jaccard, P. (1901). 73

[JAIN and YU, 1998] JAIN, A. K. and YU, B. (1998). Automatic text location in images and video frames. *Pattern Recognition*, 31(12) :2055 – 2076. 91

[Jolion, 2000] Jolion, J.-M. (2000). *Les systèmes de vision*. Hermès science publications. 9

[Jones et al., 2010] Jones, D. H., Powell, A., Bouganis, C.-S., and Cheung, P. Y. K. (2010). *A Salient Region Detector for GPU Using a Cellular Automata Architecture*, pages 501–508. Springer Berlin Heidelberg, Berlin, Heidelberg. 94

[Jorion, 1997] Jorion, P. (1997). Ce que penrose dit vraiment. pages 9–13. 5

[Ke and Sukthankar, 2004] Ke, Y. and Sukthankar, R. (2004). Pca-sift : a more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–506–II–513 Vol.2. 25

[Kim, 1996] Kim, H.-K. (1996). Efficient automatic text location method and content-based indexing and structuring of video database. *Journal of Visual Communication and Image Representation*, 7(4) :336 – 344. 91

[Kim et al., 2002] Kim, I.-K., Jung, D.-W., and Park, R.-H. (2002). Document image binarization based on topographic analysis using a water flow model. *Pattern Recognition*, 35(1) :265 – 277. Shape representation and similarity for image databases. 93

[Kindermann and Snell, 1980] Kindermann, R. and Snell, J. L. (1980). *Markov random fields and their applications*. American Mathematical Society, Providence. 90

[Kise et al., 1998] Kise, K., Sato, A., and Iwata, M. (1998). Segmentation of page images using the area voronoi diagram. *Computer Vision and Image Understanding*, 70(3) :370 – 382. 88

[Kottman, 2011] Kottman, M. (2011). The color-brief feature descriptor. 32

[Lelore and Bouchara, 2013] Lelore, T. and Bouchara, F. (2013). Fair : A fast algorithm for document image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8) :2039–2048. 94

[Leutenegger et al., 2011] Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk : Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2548–2555, Washington, DC, USA. IEEE Computer Society. iii, 20, 34

[Levi and Hassner, 2016] Levi, G. and Hassner, T. (2016). LATCH : learned arrangements of three patch codes. In *Winter Conference on Applications of Computer Vision (WACV)*. IEEE. iii, 39, 40

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110. ii, 18, 23, 24, 54

[Mair et al., 2010] Mair, E., Hager, G. D., Burschka, D., Suppa, M., and Hirzinger, G. (2010). Adaptive and generic corner detection based on the accelerated segment test. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*. ii, 16, 18

[Martins et al., 2012] Martins, P., Carvalho, P., and Gatta, C. (2012). Context aware keypoint extraction for robust image representation. In *Proceedings of the British Machine Vision Conference*, pages 100.1–100.12. BMVA Press. 49

[Matas et al., 2002] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press. doi :10.5244/C.16.36. 86

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10) :1615–1630. iii, 26

[Mok et al., 2011] Mok, S. J., Jung, K., Ko, D. W., Lee, S. H., and Choi, B.-U. (2011). Serp : Surf enhancer for repeated pattern. In *Proceedings of the 7th International Conference on Advances in Visual Computing - Volume Part II*, ISVC'11, pages 578–587, Berlin, Heidelberg. Springer-Verlag. 63

[Mortensen et al., 2005] Mortensen, E. N., Deng, H., and Shapiro, L. (2005). A sift descriptor with global context. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 184–190, Washington, DC, USA. IEEE Computer Society. iv, 51

[Nagy, 1984] Nagy, G. (1984). Hierarchical representation of optically scanned documents. In *Proc. 7th Int. Conf. Patt. Recogn.*, pages 347–349. 90

[Nakai et al., 2006] Nakai, T., Kise, K., and Iwamura, M. (2006). *Use of Affine Invariants in Locally Likely Arrangement Hashing for Camera-Based Document Image Retrieval*, pages 541–552. Springer Berlin Heidelberg, Berlin, Heidelberg. 85

[Niblack, 1985] Niblack, W. (1985). *An Introduction to Digital Image Processing*. Strandberg Publishing Company, Birkeroed, Denmark, Denmark. 92

[Nicolas et al., 2006] Nicolas, S., Paquet, T., and Heutte, L. (2006). Extraction de la structure de documents manuscrits complexes à l'aide de champs markoviens. In *Actes du 9ème Colloque International Francophone sur l'Ecrit et le Document*, pages 13–18. SDN06. 90

[O'Gorman, 1993] O'Gorman, L. (1993). The document spectrum for page layout analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11) :1162–1173. 88

[Ojala et al., 1996] Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1) :51–59. 28

[Ojala et al., 2002] Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :971–987. 29

[Okamoto and Takahashi, 1993] Okamoto, M. and Takahashi, M. (1993). A hybrid page segmentation method. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 743–746. 88

[Otsu, 1979] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1) :62–66. 91

[Parzen, 1962] Parzen, E. (1962). On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3) :1065–1076. 57

[Pearson, 1901] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2 :559–572. 25

[Petric,] Petric, S. Image effects with cellular automata. 94

[Pratikakis et al., 2011] Pratikakis, I., Gatos, B., and Ntirogiannis, K. (2011). Icdar 2011 document image binarization contest (dibco 2011). In *2011 International Conference on Document Analysis and Recognition*, pages 1506–1510. 94

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1) :81–106. 16

[Rafler, 2011] Rafler, S. (2011). Generalization of conway's "game of life" to a continuous domain - smoothlife. 95

[Ramírez-Ortegón et al., 2010] Ramírez-Ortegón, M. A., Tapia, E., Ramírez-Ramírez, L. L., Rojas, R., and Cuevas, E. (2010). Transition pixel : A concept for binarization based on edge detection and gray-intensity histograms. *Pattern Recognition*, 43(4) :1233–1243. 93

[Rosenblatt, 1956] Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.*, 27(3) :832–837. 57

[Rosin, 1999] Rosin, P. (1999). Measuring corner properties. In *Computer Vision & Image Understanding, Vol.73, No.2*, pages 291–307. 20

[Rosin, 2005] Rosin, P. L. (2005). *Training Cellular Automata for Image Processing*, pages 195–204. Springer Berlin Heidelberg, Berlin, Heidelberg. 94

[Rosten and Drummond, 2006] Rosten, E. and Drummond, T. (2006). *Machine Learning for High-Speed Corner Detection*, pages 430–443. Springer Berlin Heidelberg, Berlin, Heidelberg. ii, 14, 15

[Rublee et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb : An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA. IEEE Computer Society. iii, 20, 32, 33, 34

[Rusiñol et al., 2015] Rusiñol, M., Chazalon, J., Ogier, J. M., and Lladós, J. (2015). A comparative study of local detectors and descriptors for mobile document classification. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 596–600. 72, 84

[Rusiñol and Lladós, 2009] Rusiñol, M. and Lladós, J. (2009). Logo spotting by a bag-of-words approach for document categorization. In *2009 10th International Conference on Document Analysis and Recognition*, pages 111–115. 53

[Saha and Démoulin, 2012] Saha, S. and Démoulin, V. (2012). Aloha : An efficient binary descriptor based on haar features. In *2012 19th IEEE International Conference on Image Processing*, pages 2345–2348. 36

[Sattler et al., 2009] Sattler, T., Leibe, B., and Kobbelt, L. (2009). Scramsac : Improving ransac's efficiency with a spatial consistency filter. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2090–2097. 55

[Sauvola et al., 1997] Sauvola, J., Seppanen, T., Haapakoski, S., and Pietikainen, M. (1997). Adaptive document binarization. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, volume 1, pages 147–152 vol.1. 92

[Schapire, 1990] Schapire, R. E. (1990). The strength of weak learnability. 42

[Shafait et al., 2008] Shafait, F., Keysers, D., and Breuel, T. M. (2008). Efficient implementation of local adaptive thresholding techniques using integral images. *DRR*, 6815 :681510. 93

[Shao and Gool, 2003] Shao, T. S. H. and Gool, L. V. (2003). Zubud-zurich buildings database for image based recognition, technical report no. 260. 61

[Skellam, 1946] Skellam, J. G. (1946). The frequency distribution of the difference between two poisson variates belonging to different populations. *J. Royal Statist. Soc.*, 109 :296. 60

[Smith and Brady, 1997] Smith, S. M. and Brady, J. M. (1997). Susan—a new approach to low level image processing. *International Journal of Computer Vision*, 23(1) :45–78. 13

[Strouthopoulos et al., 1997] Strouthopoulos, C., Papamarkos, N., and Chamzas, C. (1997). Identification of text-only areas in mixed-type documents. *Engineering Applications of Artificial Intelligence*, 10(4) :387 – 401. 88

[T. Trzcinski and Fua, 2012] T. Trzcinski, M. Christoudias, V. L. and Fua, P. (2012). Learning Image Descriptors with the Boosting-Trick. In *NIPS*. 42

[Trier and Taxt, 1995] Trier, O. D. and Taxt, T. (1995). Improvement of "integrated function algorithm" for binarization of document images. *Pattern Recognition Letters*, 16 :277–283. 94

[Trzcinski and Lepetit, 2012] Trzcinski, T. and Lepetit, V. (2012). Efficient Discriminative Projections for Compact Binary Descriptors. In *European Conference on Computer Vision*. iii, 41, 42

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–511–I–518 vol.1. 36

[Wieser and Pinz, 1993] Wieser, J. and Pinz, A. (1993). Layout and analysis : Finding text, titles, and photos in digital images of newspaper pages. In *2nd International Conference Document Analysis and Recognition, ICDAR '93, October 20-22, 1993, Tsukuba City, Japan*, pages 774–777. 90

[Wolf and Doermann, 2002] Wolf, C. and Doermann, D. (2002). Binarization of low quality text using a markov random field model. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 160–163. IEEE. 94

[Wong et al., 1982] Wong, K. Y., Casey, R. G., and Wahl, F. M. (1982). Document analysis system. *IBM J. Res. Dev.*, 26(6) :647–656. v, 88, 89

[Zagoris and Pratikakis, 2012] Zagoris, K. and Pratikakis, I. (2012). *Scene Text Detection on Images Using Cellular Automata*, pages 514–523. Springer Berlin Heidelberg, Berlin, Heidelberg. 94

[Zhang et al., 2012] Zhang, L., Zhou, Z., and Li, H. (2012). Binary gabor pattern : An efficient and robust descriptor for texture classification. In *2012 19th IEEE International Conference on Image Processing*, pages 81–84. iii, 38

[Zhang et al., 2013] Zhang, S., Tian, Q., Lu, K., Huang, Q., and Gao, W. (2013). Edge-sift : Discriminative binary descriptor for scalable partial-duplicate mobile search. *IEEE Transactions on Image Processing*, 22(7) :2889–2902. iii, 43, 44

Résumé

Ce travail s'inscrit dans une tentative de liaison entre la communauté classique de la Vision par ordinateur et la communauté du traitement d'images de documents, analyse et reconnaissance (DAR). Plus particulièrement, nous abordons la question des détecteurs de points d'intérêts et des descripteurs locaux dans une image. Ceux-ci ayant été conçus pour des images issues du monde réel, ils ne sont pas adaptés aux problématiques issues du document dont les images présentent des caractéristiques visuelles différentes. Notre approche se base sur la résolution du problème de la confusion entre les descripteurs, ceux-ci perdant leur pouvoir discriminant. Notre principale contribution est un algorithme de réduction de la confusion potentiellement présente dans un ensemble de vecteurs caractéristiques d'une même image, ceci par une approche probabiliste en filtrant les vecteurs fortement confusifs. Une telle conception nous permet d'appliquer des algorithmes d'extractions de descripteurs sans avoir à les modifier ce qui constitue une passerelle entre ces deux mondes.

Summary

This work tries to establish a bridge between the field of classical computer vision and document analysis and recognition. Specificaly, we tackle the issue of keypoints detection and associated local features computation in the image. These are not suitable for document images since they were designed for real-world images which have different visual characteristic. Our approach is based on resolving the issue of reducing the confusion between feature vectors since they usually lose their discriminant power with document images. Our main contribution is an algorithm reducing the confusion between local features by filtering the ones which present a high confusing risk. We are tackling this by using tools from probability theory. Such a method allows us to apply features extraction algorithms without having to modify them, thus establishing a bridge between these two worlds.