Localisation visuelle multimodale visible/infrarouge pour la navigation autonome

Fabien Bonardi

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité Informatique

Préparée au sein de Normandie Université

Localisation visuelle multimodale visible/infrarouge pour la navigation autonome

Présentée et soutenue par
Fabien BONARDI

Thèse soutenue publiquement le 23 novembre 2017 devant le jury composé de		
Mme Luce MORIN	Professeur des universités, INSA de Rennes	Rapporteure
M. Claude PÉGARD	Professeur des universités, Université de Picardie Jules Verne	Rapporteur
M. Ludovic MACAIRE	Professeur des universités, Université Lille 1	Examinateur
Mme Samia BOUCHAFA	Professeur des universités, Université d'Évry, Paris Sud	Examinatrice
M. Rémi BOUTTEAU	Enseignant-chercheur, ESIGELEC/IRSEEM	Examinateur
M. Pascal VASSEUR	Professeur des universités, Université de Rouen	Directeur de thèse
M. Xavier SAVATIER	Enseignant-chercheur HDR, ESIGELEC/IRSEEM	Co-encadrant
Mme Samia AINOUZ	Maître de conférence, INSA de Rouen	Co-encadrante

Thèse dirigée par Pascal VASSEUR, Laboratoire LITIS, Université de Rouen

UNIVERSITÉ DE ROUEN NORMANDIE

MIIS

litis irseem

Introduction générale

Les aptitudes des robots n'ont cessé d'évoluer au cours des dernières décennies. Autrefois cantonnés en simples automates au cœur des usines du XX$^{\text{ème}}$ siècle, l'essor des disciplines liées à la robotique et l'intelligence artificielle a permis de leur confier des tâches plus complexes et en interaction avec un environnement dynamique. Plus seulement dédiés à des actions atomiques et répétitives, ils assistent et remplacent désormais l'humain dans des domaines variés : drones aériens de photographie, robots d'exploration sous-marine ou spatiale, robots domestiques d'aide à la personne et de service, *etc.* Ces agents mobiles sont toutefois encore très dépendants de décisions et d'ordres humains. Les enjeux de ces dernières années sont de les rendre d'autant plus autonomes en les affranchissant le plus possible d'un téléguidage quelconque.

On regroupe ainsi sous l'expression *navigation autonome* l'ensemble des méthodes visant à automatiser les déplacements d'un agent mobile. Il peut s'agir de robots d'exploration, de service ou même de transport, et par extension de véhicules intelligents. Le domaine de la navigation autonome recèle ainsi de situations et contextes très variés : évolution à l'intérieur de bâtiments ou en milieux extérieurs voire milieux naturels, environnements structurés et maîtrisés ou totalement inconnus, environnements dynamiques dont l'apparence évolue fortement, connaissance ou non de l'environnement *a priori*, *etc.* Nous pouvons globalement décomposer le problème de la navigation autonome en sous-tâches. Afin de réaliser un déplacement, un agent doit estimer sa position (*localisation*) dans un environnement qu'il connaît suffisamment (*cartographie*) par rapport à une destination (*planification de trajectoire*). L'exploration demeure un cas particulier de cette analyse, en effet, le robot ne possède ni de connaissance initiale de son environnement, ni de but précis autre que celui de composer une cartographie de son environnement. Les travaux présentés dans ce manuscrit se concentrent sur la problématique de la localisation en milieu extérieur, urbain, périurbain et rural. La diversité du problème s'exprime en outre par la variété des capteurs et sources d'informations disponibles. Les travaux présentés dans ce mémoire de thèse approchent la problématique de la localisation visuelle soumise à la fois à un changement de capteurs (géométrie et modalité) ainsi qu'aux changements de l'environnement à long terme, contraintes combinées encore très peu étudiées dans l'état de l'art.

Ce mémoire de thèse se divise en trois parties allant de l'état de l'art sur la navigation autonome jusqu'à la mise en pratique d'un système de localisation visuelle à long terme :

La première partie fait état des solutions employées actuellement en robotique mobile. Ces systèmes, de plus en plus complexes, font appel à une architecture liant

capteurs et actionneurs avec des algorithmes de différents niveaux d'abstraction. Nous présentons dans cette partie un état de l'art général sur la robotique mobile et la navigation en précisant les types de capteurs fréquemment utilisés. Le choix de ces capteurs et des méthodes associées sera déterminé par le contexte de navigation (intérieur ou extérieur). De même, la représentation en mémoire de l'environnement, ou cartographie, varie selon la dimension et la nature du milieu. Les recherches menées dans le cadre de cette thèse ont porté sur l'utilisation exclusive de capteurs de vision. Pour cela, nous présentons dans cette partie les méthodes de vision par ordinateur, et tout particulièrement les méthodes d'extraction et compression des informations utiles et observables dans une image pour la reconnaissance de lieu (*Visual Place Recognition*).

Dans la deuxième partie, deux nouvelles méthodes sont proposées : la première est une méthode de description globale de l'image liée aux paramètres géométriques du capteur optique ; la deuxième est une méthode de description ponctuelle conçue pour faire face à la fois aux changements d'apparence à long terme de l'environnement ainsi qu'aux changements de modalité des capteurs utilisés. Nous évaluons la répétabilité des détecteurs connus dans la littérature (GFTT, SIFT, FAST…) afin d'en choisir le plus invariant aux changements de modalité. Nous avons également présenté dans cette partie notre contribution majeure qui porte sur la phase de description et compression des données sous la forme d'un histogramme de mots visuels que nous avons nommée PHROG (Plural Histogrammes of Restricted Oriented Gradients). Les expériences menées ont été réalisées sur plusieurs bases d'images avec différentes modalités visibles et infrarouges. Certaines bases sont bien connues dans l'état de l'art (VPRiCE, EPFL) et nous avons réalisé un jeu de données à partir de caméra visible et SWIR (infrarouge proche). Les résultats obtenus démontrent une amélioration des performances de reconnaissance de scènes comparés aux méthodes de l'état de l'art.

Dans la troisième partie de ce mémoire, nous nous intéressons à la nature séquentielle des images acquises dans un contexte de navigation afin de filtrer et supprimer des estimations de localisation aberrantes. Ces erreurs ponctuelles sont inhérentes à toutes les méthodes de localisation visuelle considérant les images unes-à-unes. Les méthodes tirant part de cette séquentialité sont abordées. Les concepts d'un cadre probabiliste Bayésien sont introduits et deux applications de filtrage probabiliste appliquées à notre problématique sont synthétisées par la suite : une première solution définit un modèle de déplacement simple du robot avec un filtre d'histrogramme et la deuxième met en place un modèle plus évolué faisant appel à l'odométrie visuelle au sein d'un filtre particulaire. La méthode PHROG proposée dans la deuxième partie permet dans ce cas de réaliser l'étape de mise à jour du filtre particulaire. Ces méthodes de filtrage proposées, testées sur notre propre base de données, ont permis de

réduire les erreurs d'estimation produites par la méthode de localisation décrite dans la deuxième partie de ce manuscrit.

Le but de ce mémoire de thèse est de contribuer à la problématique de la localisation visuelle à long terme usant de capteurs variés en terme de géométrie et modalité. Nous avons démontré que les méthodes de détection les plus modernes ne sont pas forcément les plus judicieuses lorsqu'il s'agit de favoriser la répétabilité d'une modalité à l'autre. Les résultats expérimentaux obtenus dans les différentes parties de ce manuscrit ont montré une amélioration des performances de localisation lorsque le système est contraint à la fois à un changement de modalité du capteur ainsi qu'aux changements d'apparence de l'environnement à long terme. Néanmoins, des cas de divergence par rapport à la vérité terrain subsistent encore dans notre cas mais également dans toutes les méthodes de la littérature basées sur la vision seule. La problématique de la localisation visuelle à long terme, soumise en outre aux contraintes de la multimodalité, reste toujours ouverte. Nous présentons en conclusion les pistes et perspectives envisageables qui devraient permettre d'améliorer encore ces méthodes.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

CHAPITRE

1

ROBOTIQUE MOBILE ET VISION PAR ORDINATEUR

Sommaire

1.1 Robotique mobile

1.1.1 De l'automate au robot

Le cycle Perception, Réflexion, Actions et l'architecture associée

La différence principale que l'on définit entre l'automate et le robot est la capacité de ce dernier à réagir à son environnement alors que le premier ne réalise qu'une suite d'action pré-établies. Un robot est donc un système doté de capacité de perception, de décision et d'action avec un certain degré d'autonomie en fonction de l'environnement dans lequel il évolue.

Le modèle de représentation de l'environnement, souvent complexe, impose la mise en place de plusieurs couches d'abstraction. Au plus bas niveau, les compétences font appel à l'automatique et aux tâches d'asservissement des actionneurs, alors qu'à plus haut niveau, on retrouve des tâches plus complexes de localisation et planification de trajectoire par exemple. Les architectures de contrôle d'un robot varient selon la façon dont on définit les interactions entre trois fonctions majeures que sont la perception, la décision et l'action. Ces architectures peuvent être hiérarchiques, réactives ou hybrides. La structure hiérachique suppose que la représentation de l'environnement *a priori* est quasiment parfaite et nécessite peu d'informations supplémentaires au moment de l'exécution de la tâche par le robot. La structure réactive quant à elle définit un ensemble de comportements réactifs sans utiliser de modèle du monde. La structure hybride se compose d'au minimum deux niveaux. Un premier niveau qui gère des tâches de haut niveau (localisation, cartographie et planification par exemple) s'appuie sur un niveau réactif qui exécute les commandes et gère la précision du système face à un environnement inconnu ou dynamique. La plupart des méthodes actuelles, définissent ainsi un cycle entre les trois fonctions majeures à plusieurs niveaux d'abstraction.

La figure 1.1 illustre ce principe pour la tâche de la navigation autonome. La strate la plus basse est la partie réactive du système et gère l'asservissement des roues de notre robot. La boucle intermédiaire permet d'estimer l'orientation du robot dans son environnement. La couche la plus haute quant à elle fait le lien entre la localisation globale et la trajectoire de navigation à effectuer.

La cartographie

La mobilité et la planification de trajectoires suppose que le système possède une connaissance *a priori* de l'environnement qu'il doit parcourir. Cette connaissance, qui peut évoluer au cours du temps, est ce que l'on nomme une carte. La nature de la

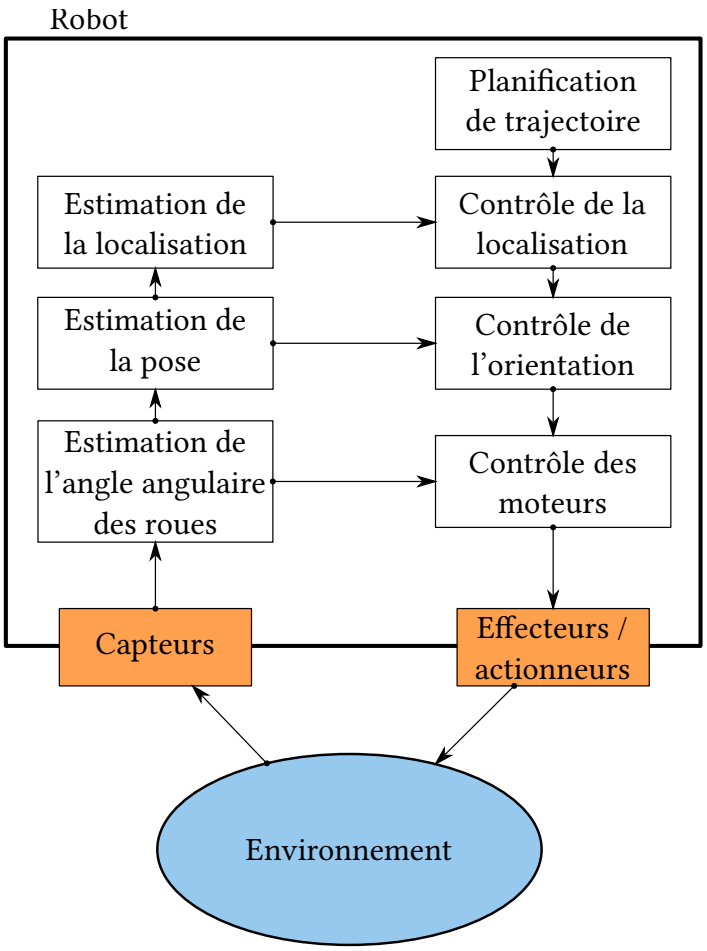


FIGURE 1.1 – Schéma du cycle perception, décision, action d'un système robotique

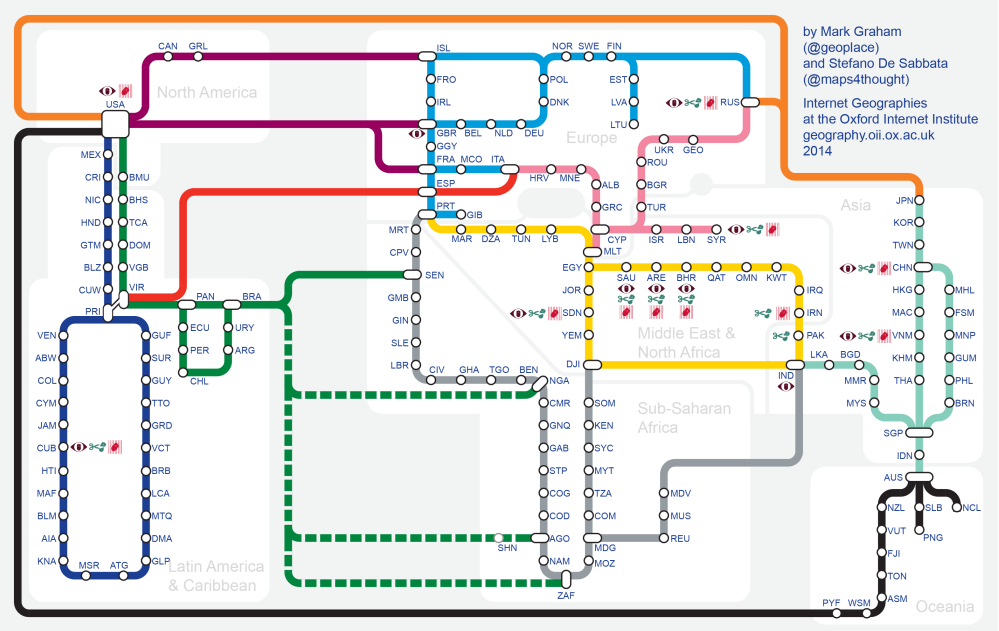


FIGURE 1.2 – Un exemple de carte topologique : visualisation des principales liaisons à fibres optiques du réseau internet à l'échelle mondiale (illustration tirée du site web geography.oii.ox.ac.uk).

carte peut prendre différentes formes en fonction des capteurs choisis pour observer l'environnement, mais aussi en fonction de la tâche à accomplir. En effet, la navigation dans un espace vaste nécessite une gestion de l'espace en mémoire efficace par exemple. De même, la carte nécessaire à une précision de navigation au centimètre ne sera pas la même que celle pour une exploration d'un environnement globalement homogène. Les méthodes dédiées à la représentation de l'environnemnet du robot sont regroupées sous le nom de «cartographie».

On distingue principalement deux type de cartes, cartes métriques et cartes topologiques. Une carte métrique définit la position d'objets selon des coordonnées exprimées dans un repère global qui peut être cartésien ou sphérique par exemple. Une carte topologique est plus abstraite : il s'agit d'un graphe dont les objets, amers ou lieux de l'environnement, composent les nœuds du graphe, et la possibilité d'un chemin navigable entre deux nœuds, les arrêtes du graphe. Un exemple très courant de carte topologique est une représentation des réseaux de télécommunications ou de transports en commun urbains tels que les stations de métro et les lignes permettant de joindre ces stations (un exemple de carte topologique est donné en figure 1.2). Dans le cadre de la navigation en robotique, on retrouve souvent une information sémantique

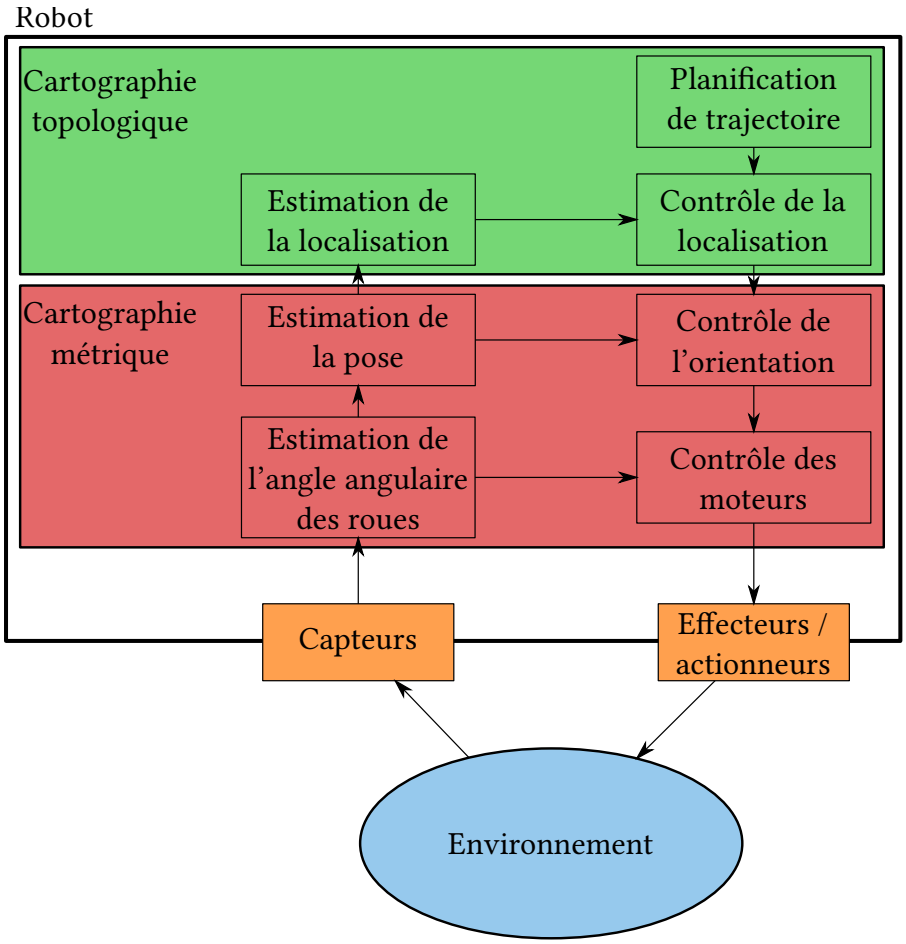


FIGURE 1.3 – Schéma du cycle perception-décision-action et lien avec le type de cartographie

liée au graphe topologique. En outre, des représentations hybrides, mêlant données métriques et abstraction topologique, existent. Cette distinction rejoint également l'architecture des systèmes que nous avons décrite précédemment (figure 1.1). En effet, les couches d'actions et perceptions les plus basses auront plutôt tendance à utiliser les données issues d'une cartographie métrique, alors que les instances de plus haut niveau auront un but défini comme un nœud du graphe (figure 1.3). La méthode de planification de trajectoire fera appel aux connexions existant entre le nœud de départ (la position courante du robot) et les nœuds intermédiaires permettant de joindre le nœud destination.

Capteurs extéroceptifs et proprioceptifs

Les informations issues de l'environnement servant à composer une carte et se localiser peuvent être acquises par une grande variété de capteurs. On distingue deux grandes familles de capteurs : les capteurs proprioceptifs et les capteurs extéroceptifs.

Un capteur extéroceptif permet d'évaluer une mesure par rapport à un point fixe de l'environnement (un amer avec des coordonnées métriques par exemple). Cette mesure, même si elle est entâchée d'erreurs (incertitudes), est absolue. Parmi les méthodes faisant usage d'informations extéroceptives en navigation robotique, on trouve la triangulation par GPS, ou encore l'estimation de position par rapport à des mires artificielles (comme des *QRcodes* présentés dans l'article [LL09]) dans un bâtiment. D'autres capteurs tels que les télémètres laser, radar, LIDAR peuvent être utilisés au cœur de méthodes analogues.

Un capteur proprioceptif mesure une information qui est relative, et le plus souvent propre à l'état antérieur du système. Il s'agit par exemple de la mesure du déplacement angulaire des roues du robot, permettant de reconstituer sa trajectoire. Il arrive souvent par contre que les roues glissent ou dérapent suivant la nature du terrain. Il s'ensuit une dérive dans l'estimation de la trajectoire qui s'accumule au cours du temps.

Les méthodes faisant appel aux capteurs proprioceptifs sont la plupart du temps plus précises. Elles sont sujettes à des erreurs et incertitudes plus faibles que pour des information extéroceptives. Néanmoins, la plupart du temps, on combine des informations proprioceptives, certe plus précises mais soumises à des dérives, avec des informations extéroceptives afin d'estimer et de compenser les dérives accumulées.

Un même capteur peut être utilisé à la fois pour des informations extéroceptives et proprioceptives. Si l'on prend l'exemple d'une caméra, reconnaître une mire visible et de taille connue dans le champ de la caméra permet destimer une position absolue de la caméra par rapport à cette mire. La même caméra pourra être utilisée pour estimer les poses entre deux de ses acquisitions successives : on parle alors d'odométrie visuelle. L'information ainsi acquise est proprioceptive. Nous reviendrons plus en détails dans la section 3.1.2 sur la technique d'odométrie visuelle.

1.1.2 La variété des environnements en robotique

Les contextes d'usage des robots font que les environnements parcourus peuvent être de type intérieur ou extérieur.

La navigation d'un robot en milieu intérieur s'exerce dans un contexte industriel (ex. : manutention dans un entrepôt) ou domestique (ex. : aide à la personne, robots compagnons et de divertissement). Ces environnements sont plutôt maîtrisés en terme d'éclairage constant et d'espaces navigables. Ils sont par contre source d'aliasing,

c'est-à-dire que les perceptions que le système reçoit de son environnement pour plusieurs localisations peuvent être très proches voire identiques (les deux positions se confondent). Pour éviter cela, il est plus aisé d'augmenter l'environnement de repères uniques qui feront office d'amers.

La navigation d'un robot en milieu extérieur quant à elle s'exerce souvent dans des contextes exploratoires (robotique sous-marine, photographie aérienne) qui sont des situations difficilement accessibles à l'homme ou des situations très contraintes comme la navigation urbaine. Dans ce dernier cas, le système se trouve plongé au cœur d'un environnement difficilement prévisible. Le système est confronté à des éléments dynamiques à court-terme tels que les véhicules (voitures et camions) et autres usagers de la route (piétons et cyclistes) ou obstacles (animaux, végétation). Le système doit en outre être capable d'interagir avec des éléments artificiels issus de l'infrastructure routière (feux tricolores par exemple). D'autres éléments présentent des variations de l'environnement à long terme. Il s'agit des variations d'illumination de la scène perçue à différents moments de la journée et suivant des conditions météorologiques variables. Les saisons apportent également des changements d'apparence sur la végétion ou dégradent les conditions de visibilité (pluie, brouillard, neige).

Degré d'autonomie et véhicule (critères NHTSA et SIA)

Comme les systèmes d'aide à la conduite délèguent de plus en plus d'actions à la machine, la navigation d'un véhicule autonome s'apparente aux tâches et contraintes rencontrées en robotique. De plus, la navigation d'un véhicule sur une infrastructure routière est particulièrement critique. En effet, en plus d'être confronté à un environnement particulièrement dynamique et hétérogène, les vitesses des acteurs en jeu sont élevées et les distances de perception requises sont plus importantes. De ce fait, des conséquences dramatiques peuvent émerger d'une défaillance du véhicule. La conception et le fonctionnement de tels systèmes doivent donc être robustes et précis. Pour cela une législation et des codifications ont été définies par plusieurs institutions comme le NHTSA (National Highway Traffic Safety Administration) et la SIA (Société des ingénieurs de l'automobile) qui définissent 5 niveaux d'automatisation allant de la conduite sans automatisation (niveau 0) à un véhicule totalement autonome (niveau 4).

1.2 Vision par ordinateur

On dénomme par *vision par ordinateur* une des disciplines majeures de l'*intelligence artificielle* contemporaine. Le terme *intelligence artificielle* inclut l'ensemble

des méthodes et techniques visant à reproduire, ou du moins imiter, des aptitudes propres à des espèces évoluées, et tout particulièrement des facultés humaines : supervision et prise de décision, conversation, compréhension de langages naturels et traduction, apprentissage automatique, *etc.* On distingue communément une forme d'intelligence artificielle comme étant faible ou forte : tandis que l'on regroupe sous le nom *intelligence artificielle faible* des processus non-sensibles qui se concentrent sur une tâche précise, aux frontières délimitées et dont les solutions pragmatiques se multiplient d'ores et déjà dans notre société, le concept d'*intelligence artificielle forte* définit une machine généraliste qui serait dotée d'une conscience, d'un esprit et capable de ressentir des émotions. Cette dernière relève encore de l'imaginaire mais suscite de nombreuses réflexions et controverses notamment au travers du courant de pensée *transhumaniste* et de l'idée de *singularité technologique*.

La vision par ordinateur vise donc à imiter des facultés que nous possédons grâce à l'un de nos cinq sens : la vue. Les recherches en vision par ordinateur entreprises se focalisent donc sur des images que l'on analyse afin d'extraire l'information utile à une application particulière choisie. Il peut s'agir de reconnaissance de forme et détection dans une image (reconnaissance d'objets ou de caractères typographiques, identification faciale ou reconnaissance par l'iris par exemple) ou de *photogrammétrie*, c'est-à-dire de méthodes permettant de mesurer des dimensions physiques observées par le biais d'images prises selon différents points de vues. On trouve parmi ces approches la *stéréovision* qui permet le calcul d'une image de profondeur pour deux images aux champs de vue proches, ainsi que le calcul de nuages de points 3D par *Structure From Motion* permettant d'estimer une représentation tridimensionnelle de l'environnement observé.

PHOTOGRAMMÉTRIE

STÉRÉOVISION

STRUCTURE FROM MOTION

Les dispositifs permettant de projeter l'image d'une scène observée sur un plan sont très anciens : Aristote par exemple au IV$^{\text{ème}}$ siècle avant J-C, décrit le principe de la *camera obscura* (chambre noire), ou *sténopé* qui donne son nom au modèle géométrique présenté dans la section 1.2.1. Des outils analogues ont été étudiés, agrémentés de lentilles optiques et developpés par la suite, notamment à l'époque de la Renaissance afin d'assister les peintres dans l'exécution des proportions et lois de la perspective. Il faudra néanmoins attendre le XIX$^{\text{ème}}$ siècle pour disposer de solutions techniques permettant de figer directement les images projetées autrefois éphémères : le *Point de vue du Gras* réalisé par Joseph Nicéphore Niépce en 1826 (figure 1.4) est ainsi passé à la postérité comme première expérience concluante de fixation permanente d'une image naturelle. Cette première photographie a tout de même nécessité un temps de pose de plusieurs heures. Louis Daguerre, collaborera par la suite avec Niépce et améliorera sensiblement le processus chimique de fixation des images en réduisant le temps de pose nécessaire à quelques minutes seulement. Son invention, nommé

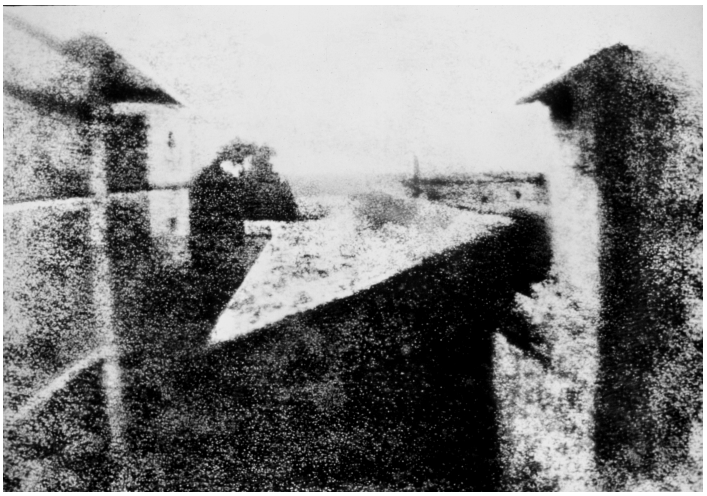


FIGURE 1.4 – *Point de vue du Gras*, par Joseph Nicéphore Niépce en 1826, première fixation photographique réussie.

le *daguerréotype*, deviendra le premier dispositif photographique commercialisé et rencontrera un grand succès.

Les progrès technologiques successifs bénéficiant à la photographie ont suivi plusieurs axes :

— l'amélioration des processus photochimiques ont permis de réduire le temps de pose nécessaire, améliorer la stabilité et reproductibilité du tirage en évitant à l'utilisateur de gérer des manipulations chimiques trop complexes
— l'amélioration des dispositifs optiques, lentilles, objectifs et boîtiers afin d'augmenter la quantité de lumière entrante, réduisant ainsi le temps de pose, tout en proposant des systèmes toujours plus légers et moins chers

La photographie bascule dans l'ère du numérique au XXI$^{\text{ème}}$ siècle : la surface sensible révélée par un processus photochimique est remplacée par un capteur électronique photosensible transformant la lumière perçue en signal électrique. L'image naturelle est alors projetée sur une matrice de pixels et se trouve ainsi discrétisée selon 3 dimensions : 2 dimensions spatiales, la hauteur et la largeur du capteur, et une dimension quantifiant l'intensité (quantité de lumière perçue) par pixel.

Une vue globale de la chaîne de traitement impliquée dans un capteur image numérique est donnée en figure 1.5. Elle peut être décomposée en trois étapes :

— le dispositif optique et physique projette l'image naturelle sur un plan et selon un temps de pose choisi. Les transformations géométriques liant objets et leur projection sur le plan image sont abordées dans la section 1.2.1 ;

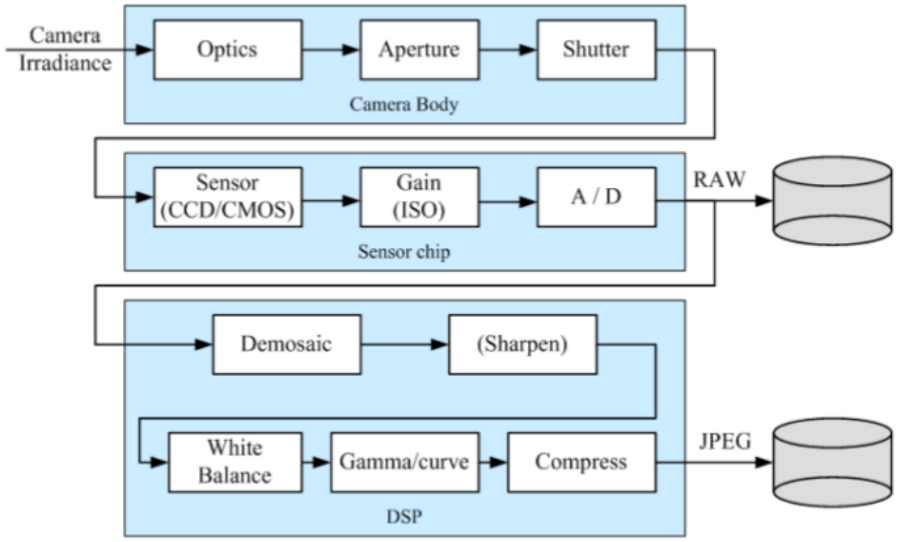


FIGURE 1.5 – Vue globale de la chaîne de traitement d'un capteur image (illustration tirée de l'ouvrage [Sze10]).

— la chaîne de traitements électronique met en forme le signal électrique issu du capteur et le quantifie. Ces processus sont explicités dans les sections 1.2.1 et 1.2.1 ;
— la chaîne de traitement logicielle, le plus souvent implémentée sur DSP embarqué, réalise dématriçage, amélioration de la netteté, réglages colorimétriques et éventuelle compression afin d'optimiser l'espace utilisé en mémoire par le rendu final des images. Ces traitements et corrections sont abordés dans la section 1.2.2.

1.2.1 Généralités et modélisation d'un capteur optique

Un capteur image est un système qui met en œuvre plusieurs modèles issus de différents domaines allant de l'optique géométrique au traitement du signal en passant par l'optique ondulatoire. Selon les besoins, nous pouvons trouver différents types de capteurs (CCD, CMOS, *etc*) dotés de technologies multiples qui ont chacun leurs avantages et inconvénients [HL07]. Nous nous concentrons dans un premier temps sur le modèle de projection utilisé liant objets physiques observés et leur image sur le plan du capteur.

Optique géométrique

Le modèle géométrique de projection d'un capteur image le plus répandu se nomme *modèle du sténopé* (ou *pinhole* en anglais). Il est présenté en détails dans l'ouvrage [HZ03] et une représentation graphique de ce modèle est donnée en figure 1.6. Il modélise une caméra perspective plongée dans un espace euclidien à 3 dimensions $\mathbb{R}^3$ par un *plan image* correspondant au capteur photosensible et un *centre de projection*, ou *centre optique* que l'on nomme **C**. On définit ensuite l'*axe optique* comme étant la droite orthogonale au plan image et passant par le centre optique : on nomme alors *point principal* **p** l'intersection de l'axe optique avec le plan image. La distance entre le centre optique et le plan image est nommée *distance focale* f. Il est d'usage de définir plusieurs repères :

Modèle du sténopé

- un repère *monde* R_m à 3 dimensions $(X_{R_m}, Y_{R_m}, Z_{R_m})$ quelconque dans lequel sont définis les objets de la scène naturelle observée ; Repère monde
- un repère caméra R_c à 3 dimensions $(X_{R_c}, Y_{R_c}, Z_{R_c})$, centré en **C** et dont la direction Z_{R_c} est orienté selon l'axe optique ;
- un repère *image* R_i à deux dimensions (x_{R_i}, y_{R_i}) dans le plan image centré selon le point principal ;
- un repère *pixellique* R_p à deux dimensions (u_{R_p}, v_{R_p}) dans le plan image et correspondant aux coordonnées pixelliques de l'image numérique acquise. Repère pixellique

Afin de simplifier les développements, on définit habituellement les repères R_c, R_i et R_p de sorte que u_{R_p}, x_{R_i} et X_{R_c} aient la même direction.

Soit un point **X** de l'espace exprimé dans le repère monde par les coordonnées homogènes $(X_m, Y_m, Z_m, 1)^T$. La projection **x** du point **X** est l'intersection du rayon optique (**CX**) avec le plan image. Comme le repère monde est arbitraire, on nomme **T** la transformation rigide permettant le changement de repère R_m vers R_c. Cette transformation rigide peut se décomposer sous la forme suivante :

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \mathbf{T} \begin{pmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0_{1\times 3} & 1 \end{pmatrix} \begin{pmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{pmatrix} = \mathbf{R} \begin{pmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{pmatrix} + \mathbf{t} \quad (1.1)$$

avec **R** la matrice de rotation entre le repère monde et le repère caméra et **t** le vecteur de translation entre ces deux repères. Ces paramètres sont dénommés *paramètres extrinsèques* de la caméra modélisée. Paramètres extrinsèques

On exprime ensuite la transformation permettant de passer du repère caméra R_c au repère image R_i. Il s'agit d'une projection perspective d'un espace euclidien à 3 dimensions $\mathbb{R}^3$ dans un espace euclidien à 2 dimensions $\mathbb{R}^2$:

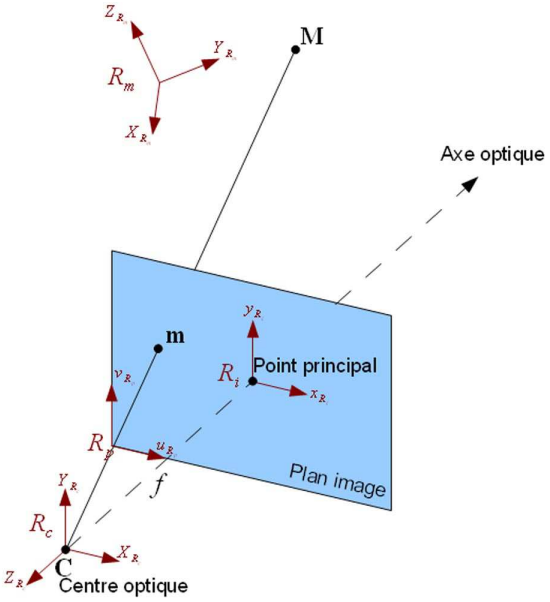


FIGURE 1.6 – Réprésentation du modèle sténopé (illustration tirée de la thèse [Bou10]).

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \propto \mathbf{P} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{1.2}$$

avec $(x, y, 1)^T$ les coordonnées du point dans le repère image.

On définit enfin la transformation affine permettant d'exprimer les coordonnées $(u, v, 1)$ du point dans le repère pixellique :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{A} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} k_u & -k_u/\cos\theta & u_0 \\ 0 & k_v/\sin\theta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{1.3}$$

avec k_u et k_v le nombre de pixels par unité de longueur selon les axes u_{R_p} et v_{R_p} ($k_v = k_u$ si les pixels sont carrés), θ l'angle d'obliquité des lignes de pixels successives et u_0 et v_0 les coordonnées pixelliques du point principal.

On considère en général que les lignes et colonnes du capteur matriciel sont orthogonales si bien que la matrice $\mathbf{A}$ se simplifie ainsi :

$$\mathbf{A} = \begin{pmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1.4}$$

On définit la matrice **K** comme le produit des matrices **A** et **P** :

$$\mathbf{K} = \mathbf{AP} = \begin{pmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.5}$$

avec $f_u = fk_u$ et $f_v = fk_v$. On nomme *paramètres intrinsèques* les paramètres de la matrice **K**. Cette matrice inclut les grandeurs physiques de la caméra : la focal exprimée en pixels dans les directions u_{R_p} et v_{R_p} ainsi que les coordonnées du point principal. Si on ne néglige pas le paramètre d'oblicité θ, la matrice **K** a la forme suivante :

Paramètres intrinsèques

$$\mathbf{K} = \begin{pmatrix} f_u & s & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.6}$$

avec $f_u = fk_u$, $f_v = fk_v / \sin\theta$ et $s = -k_u / \cos\theta$ le paramètre de biais, ou *skew* en anglais.

En résumé, la projection **x** du point **X** de l'espace est donné par la relation :

$$\mathbf{x} \propto \mathbf{P}_{proj}\mathbf{X} = \mathbf{KTX} \tag{1.7}$$

avec $\mathbf{P}_{proj}$ nommée la *matrice de projection.*

Matrice de projection

Photométrie, optique ondulatoire et réponse spectrale

Nous avons vu dans la partie précédente par quelles règles géométriques nous pouvons lier la position d'éléments ponctuels dans l'environnement tridimensionnel observé à leur projection dans le plan image du capteur. Ceci détermine donc comment sont acquises les informations selon les deux dimensions spatiales d'une image. La troisième dimension évoquée en introduction de ce chapitre est l'*intensité* quantifiée par chaque pixel. Cette intensité s'exprime sur une palette de nuances de gris pour un capteur monocanal. Dans le cas d'images couleurs, nous disposons de deux canaux supplémentaires, nous permettant de profiter de trois images d'intensité pour les trois couleurs primaires rouge, vert et bleu. Nous donnons quelques détails supplémentaires sur la manière d'acquérir des images couleurs dans la section 1.2.1.

Intensité

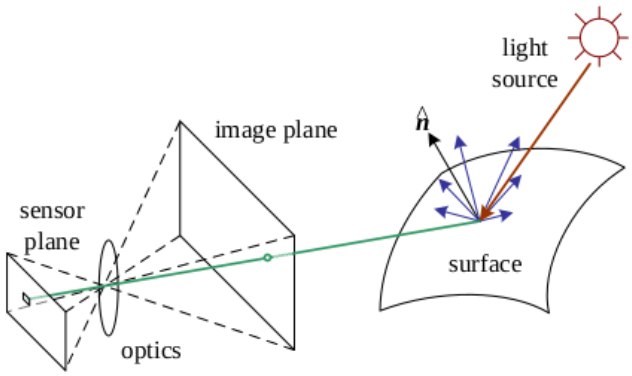


FIGURE 1.7 – Représentation du processus photométrique de formation d'une image. La lumière est émise par une ou plusieurs sources et réfléchie partiellement par un objet de la scène en direction du dispositif photographique (illustration tirée du livre [Sze10]).

Les valeurs d'intensité sont liées aux propriétés ondulatoires de la lumière et à son spectre.

Les valeurs d'intensité échantillonnées dépendent de plusieurs facteurs :

- de la ou des sources de lumière de l'environnement observé. Ces sources de lumière peuvent être *ponctuelles* ou *étendues*, *blanches*, *monochromatiques* ou *polychromatiques* et d'intensités variables ;
- de la géométrie et des propriétés de surface des objets observés, en particulier les propriétés de *réflexion*, *diffusion* et *absorption* de la lumière ;
- des caractéristiques du système optique (objectifs et qualités des lentilles) utilisé par le procédé imageur ;
- des propriétés du capteur photosensible, et tout particulièrement sa *réponse spectrale*.

Le processus photométrique de formation d'une image est schématisé en figure 1.7. Une source émet une lumière dont l'intensité est fonction d'un spectre électromagnétique et d'une direction d'émission que l'on note $L(\mathbf{v}, \lambda)$. Ce rayonnement lumineux atteint la surface d'un objet qui va le diffuser et le réfléchir en fonction de ses propriétés. Le formalisme le plus général pour décrire ces propriétés est une *fonction de distribution de la réflectivité bidirectionnelle* ou BRDF (*bidirectional reflectance distribution function* en anglais) schématisé en figure 1.8 et que l'on note $f_r(\theta_i, \phi_i, \theta_r, \phi_r, \lambda)$.

BRDF : BIDIRECTIONAL REFLECTANCE DISTRIBUTION FUNCTION

La quantité de lumière $L_r(\mathbf{v}_r, \lambda)$ émise par un point de la surface dans une direction $\mathbf{v}_r$ en fonction des sources $L_i(\mathbf{v}_i, \lambda)$ s'exprime par la formule suivante :

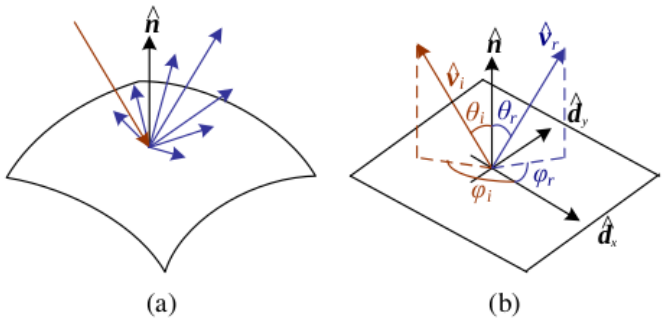


FIGURE 1.8 – Bidirectional reflectance distribution function (BRDF) caractérisée par la direction des rayons de lumière incidents $\mathbf{v}_i$ et réfléchis $\mathbf{v}_r$ et les angles qu'ils forment avec la surface tangente à l'objet au point de réflexion (illustration tirée du livre [Sze10]).

$$L_r(\mathbf{v}_r, \lambda) = \int L_i(\mathbf{v}_i, \lambda) f_r(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) \cos^+ \theta_i d\mathbf{v}_i \tag{1.8}$$

avec $\cos^+ \theta_i = \max(0, \cos \theta_i)$. Plusieurs hypothèses ont été proposées dans la littérature pour simplifier ce modèle. Certains modèles en particuliers sont présentés et décrits dans l'ouvrage [Sze10].

Les différentes parties d'un système imageur ont en outre des propriétés physiques qui ont un impact sur leur *transmittance optique* qui est fonction de la longueur d'onde λ du rayonnement lumineux perçu. Cela dépend des matériaux utilisés pour les verres optiques de l'objectif ainsi que des éventuels filtres ajoutés dans la chaîne d'acquisition optique (filtres UV et infrarouge ou polariseurs par exemple). La plupart des appareils photo du commerce sont équipés d'un filtre infrarouge placé sur le capteur. La figure 1.9 donne par exemple la comparaison entre la réponse spectrale d'un appareil photo numérique non modifié et celle de ce même appareil auquel on a retiré le filtre infrarouge. Enfin, le capteur photoélectronique a une réponse fonction de la longueur d'onde λ des photons incidents que l'on nomme *efficacité quantique*.

TRANSMITTANCE OPTIQUE

EFFICACITÉ QUANTIQUE

Sensibilité spectrale et capteurs basés vision actifs et passifs

Il existe une grande variété de capteurs imageurs qui ont chacun leurs avantages et inconvénients et dont le choix dépend principalement de l'application visée. Nous avons vu dans la partie précédente que les capteurs avaient une sensibilité spectrale variant d'un système à l'autre. Les capteurs images couleurs ont trois canaux, permettant d'acquérir des valeurs d'intensité pour trois couleurs primaires composant le

FIGURE 1.9 – Réponse spectrale d'un appareil photo numérique avec et sans filtre infrarouge (illustration tirée de la page web www.astrosurf.com/luxorion/apn-ir-uv.htm).

spectre visible (capteur RGB). Certains capteurs se focalisent sur des longueurs d'onde du domaine infrarouge : cette partie du spectre permet entre autre de réaliser des dispositifs de vision nocturne, sensibles aux rayonnements émis par la chaleur de corps physiques qui sont invisibles à l'œil nu. D'autres capteurs plus complexes possèdent de nombreux canaux avec chacun une bande de sensibilité spectrale fine. On parle de caméras *multispectrales* ou *hyperspectrales* en fonction du nombre de bandes spectrales disponibles. Ces caméras ont des applications allant de l'imagerie par satellite (pour des analyses géologiques ou agricoles par exemple) à la caractérisation de matériaux par des méthodes non-intrusives (analyse d'œuvres d'art anciennes notamment).

IMAGERIE MULTISPECTRALE

La réponse spectrale d'un système dépend donc de la sensibilité spectrale du capteur mais aussi de la composition de la lumière renvoyée par la scène observée. Ainsi un capteur basé vision passif ne fait que recevoir la lumière ambiante d'une scène. La variété du spectre lumineux reçu sera donc celle de la lumière et autres rayonnements naturels (ou d'éventuelles sources de lumière artificielle). Celle-ci n'est pas maîtrisée. D'autres systèmes à l'opposé, dits actifs, émettent leurs propres sources de lumière selon un spectre déterminé. Certains dispositifs de vision nocture émettent une lumière invisible à l'œil nu par exemple.

D'autres systèmes de vision actifs utilisent l'émission de lumière pour obtenir une information plus riche. En effet, en projetant un motif tel que des lignes parallèles, un algorithme de traitement pourra estimer la distance entre l'objet observé et la source du motif projeté (souvent la caméra elle-même). On parle d'estimation de la profondeur par émission de *lumière structurée*. On peut aussi citer les caméras «Time-of-Flight» qui émettent un flash de lumière invisible à l'œil nu, calcule le temps de parcours de la lumière émise par ce flash et en déduisent une image de profondeur de la scène observée.

LUMIÈRE STRUCTURÉE

Par analogie avec ces méthodes, on trouve parfois dans la littérature des articles qui qualifient et traitent la réponse d'un LIDAR multi-nappes comme des images. En effet, en projetant des rayons laser infrarouges, les rayons réfléchis n'ont pas tous la

même intensité si bien que l'on dispose en sortie à la fois d'une carte de profondeur et d'une image d'intensités.

1.2.2 Traitements et amélioration des images issues du capteur

Les défauts optiques des capteurs image

Les différentes parties de la chaîne d'acquisition d'un capteur photographique intègrent des défauts dans la formation des images qu'ils génèrent. Contrairement au modèle mathématique du sténopé définissant une projection selon un point, un système réel est composé d'un *diaphragme* dont on peut faire varier l'ouverture. L'usage d'objectifs permet également, en plus de jouer sur le grossissement des images, de concentrer les rayons lumineux et donc de réduire le *temps de pose* nécessaire à l'acquisition d'une image. Cet avantage est contrebalancé par le fait que la *profondeur de champ* s'en trouve réduite. Il est ainsi nécessaire de définir un compromis entre temps de pose pour chaque image, sensibilité du capteur et profondeur de champ. Dans le cas d'une profondeur de champ réduite, le réglage de mise au point définira la profondeur de la scène pour laquelle les objets observés apparaîtront nets. Le dispositif optique étant composé la plupart de temps de lentilles successives, celles-ci sont à l'origine de défauts supplémentaires dans la formation des images :

Diaphragme

Temps de pose

Profondeur de champ

- *distorsions* : les courbures des lentilles, en particulier pour des objectifs de courte focale et grand angle, peuvent provoquer des déformations géométriques en courbant des lignes droites de l'environnement. Un exemple de distortion «en coussins» est donné en figure 1.10. Ces distorsions peuvent être corrigées en appliquant un modèle mathématique simple sur les images issues du capteur ; Distorsions
- *aberrations sphériques* : les courbures peuvent également provoquer une réfraction plus importante des rayons lumineux aux bords des lentilles (*cf* figure 1.11) ; Aberrations sphériques
- *aberrations chromatiques* : l'indice de réfraction des lentilles dépend de la longueur d'onde si bien que la mise au point est différente pour chaque couleur (*cf* figure 1.12). Il en résulte des contours qui apparaîssent irisés, notamment en périphérie des images ; Aberrations chromatiques
- *vignettage* : certaines optiques laissent pénétrer moins de lumière en périphérie de l'image qu'au centre. Ceci provoque un assombrissement des coins de l'image produite. Vignettage

Les défauts électroniques des capteurs image

La surface photosensible que l'on nomme communément capteur image est la partie du dispositif convertissant la lumière entrante (les photons) en signal électrique. Il

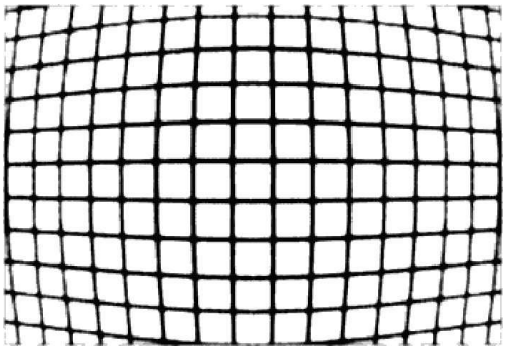


FIGURE 1.10 – Exemple de distorsion «en coussins» appliqué à une grille orthogonale (illustration tirée de la thèse [Bou10]).

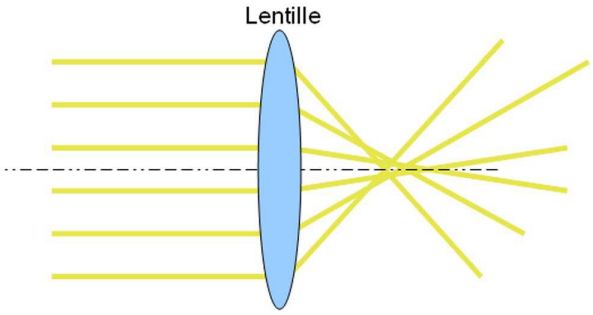


FIGURE 1.11 – Aberrations sphériques à l'origine de perte de netteté en périphérie de l'image (illustration tirée de la thèse [Bou10]).

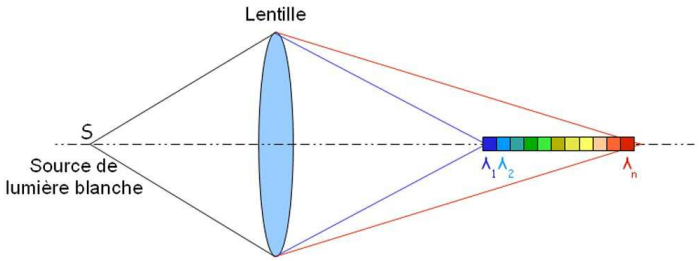


FIGURE 1.12 – Aberrations chromatiques provoquant des contours irisés en périphérie de l'image (illustration tirée de la thèse [Bou10]).

existe plusieurs technologies permettant ces applications, les plus courante étant la technologie CCD et la technologie CMOS. Ces capteurs peuvent être de sensibilité variable. Cela permet notamment d'acquérir l'image d'une scène en basse luminère et de compenser cette faible exposition par un gain d'autant plus important du signal. Cette opération est souvent mentionnée comme le réglage ISO pour les appareils photographiques grand public. Plus le gain utilisé est grand, plus l'image produite a de chances d'être bruitée.

D'autres défauts relatifs au capteur électronique peuvent être mentionnés en rapport avec la vitesse d'acquisition du système. Certains systèmes n'ont en effet pas d'obturateur physique et toutes les valeurs des photosites ne sont pas échantillonnées en même temps. Dans le cas d'un capteur effectuant une «lecture» des photosites ligne par ligne, les valeurs des photosites des premières lignes sont enregistrées (donc figées) pendant que les photosites des lignes suivantes restent sensibles à la lumière pénetrant le capteur, et ainsi aux évolutions de la scène observée. Ceci provoque des déformations de la scène ou des objets de la scène si ceux-ci sont en mouvement. On parle de *rolling shutter*. Celui-ci est plus ou moins prononcé en fonction de la vitesse de lecture successive des photosites. Une autre déformation comparable est due aux mêmes contraintes que l'on appelle «entrelacement». Les capteurs utilisant cette technique doublent la vitesse d'acquisition des images en utilisant les lignes paires pour une image puis les lignes impaires pour la suivante. Il en résulte un «effet de peigne» que certains algorithmes dits de «désentralecement» compensent en partie.

Rolling shutter

Entrelacement

Filtrage, débruitage et amélioration de la netteté

Certains systèmes, particulièrement pour le grand public, possèdent un processeur ou un circuit électronique dédié qui réalise directement des traitements sur l'image numérique issue du capteur. Il s'agit par exemple de méthodes de filtrage permettant de réduire la perception du bruit d'acquisition, en particulier dans les images prises à haute sensibilité. On trouve parmi ces méthodes de filtrage les filtres médian ou gaussien par exemple. L'application de ces filtres provoquent une perte de netteté des images (on parle aussi de «piqué») si bien que des méthodes d'augmentation de la netteté (ou *sharpening*) permettent de compenser ce manque artificiellement. On peut citer notamment la méthode du masque flou (*Unsharp masking*) qui additionne à l'image originale une copie adoucie par un filtre Gaussien et inversée, ce qui permet de donner plus de contrastes aux contours de l'image.

Sharpening

Masque flou

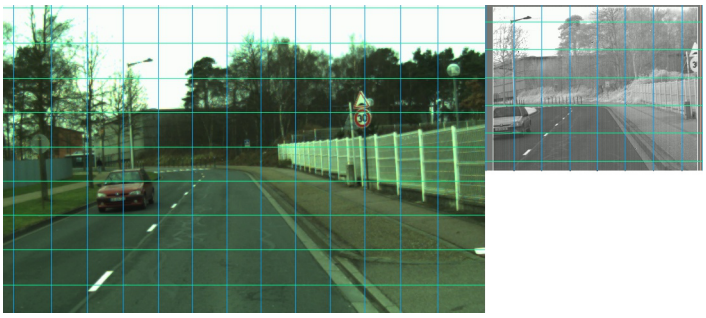


FIGURE 1.13 – Exemple d'échantillonnage de caractéristiques en grille fixe.

1.3 Extraction des informations : Primitives des images

1.3.1 «Quantification»/échantillonnage des informations de l'image

Une image brute est une quantité d'information importante en soi. La discipline de la vision par ordinateur s'attèle à définir le type d'information (critères photométriques, formes, contrastes, textures, *etc*) que l'on va extraire à l'aide de méthodes données et adaptées à l'application visée. On choisit ainsi des caractéristiques données, ou

FEATURES

features, que l'on va calculer plutôt que d'autres. Si l'on s'intéresse à une image dans sa globalité sans *a priori* sur la nature des objets ou de la scène observée, deux possibilités s'offrent à nous : la première est d'extraire les caractéristiques selon une «grille» fixe, la deuxième consiste à détecter des *points d'intérêt* comme nous le verrons dans la partie 1.3.1.

Méthodes fixes, arbitraires

La méthode la plus simple pour extraire de l'information d'une image dans sa globalité est de diviser l'image en zones arbitraires. Ces zones sont la plupart du temps des carrés de dimensions pixelliques identiques [SNP13, NSBS14] mais elles peuvent prendre d'autres formes dans le cas d'usage d'optiques *fisheye* ou omnidirectionnelle par exemple. Un exemple d'échantillonnage par grille est donné dans la figure 1.13.

Au cours des travaux de cette thèse, nous avons proposé une méthode de détermination de la grille d'échantillonnage des features liée aux paramètres intrinsèques du capteur de façon à pouvoir comparer des features extraites d'images issues de capteurs aux caractéristiques différentes. Cette méthode est présentée dans la partie 2.1.

Détection de points d'intérêt

On trouve dans la littérature deux termes distincts, *feature detection* et *feature extraction* , qui représentent selon les auteurs plus ou moins le même concept. Nous choisissons ici de distinguer les deux en précisant que la méthode de détection réalise la localisation de la caractéristique dans l'image (et détermine éventuellement sa taille, son orientation ou même sa forme dans le cas de détecteurs de blob que nous étudierons par la suite) alors que les méthodes dites d'extraction incluent aussi la phase de description de la caractéristique (c'est-à-dire l'information utile par la suite, la plupart du temps sous forme de vecteurs de scalaires ou de chaînes de bits).

Feature extraction

Point d'intérêt, région d'intérêt ou caractéristique locale

Une caractéristique locale idéale serait un point au sens géométrique, c'est-à-dire avec une localisation précise et de taille nulle. Dans les faits, les images numériques sont une représentation discrète, une quantification de la lumière émise par l'environnement projetée sur le plan du capteur. L'unité spatiale la plus petite est ainsi le pixel et il est nécessaire de considérer un voisinage local de chaque pixel à analyser pour déterminer si la zone considérée relève d'une caractéristique de type point ou non. Dans certains cas, en particulier pour des tâches de calibration ou reconstruction 3D de l'environnement qui nécessitent des positions les plus précises possibles de points, un ou différents modèles de points (une fois quantifiés par le capteur image) sont ajustés afin d'inférer une position *sub-pixellique* de l'hypothétique point. On parle ainsi de *point d'intérêt*.

Néanmoins, la plupart des applications nécessitent par la suite d'associer, et donc comparer différents points d'intérêt émanant de plusieurs images. On considère donc une *région d'intérêt* autour du point établi pour laquelle nous calculerons une description. En général, la région d'intérêt correspond au voisinage qui a été utilisé pour la phase de détection du point, mais ce n'est pas toujours le cas. De plus, afin d'obtenir une description autour de points d'intérêt invariante aux rotations, mais aussi aux transformations affines et projectives, cette région d'intérêt peut nécessiter ré-échantillonnage ou interpolation avant de procéder à l'étape de description (ces détails seront présentés par la suite).

Le terme caractéristique locale (*local feature*) fait ainsi référence à un point d'intérêt accompagné d'une région d'intérêt choisie dans son voisinage proche et sur laquelle nous calculons une description.

Qu'est-ce qu'une caractéristique locale ?

On désigne de façon générale comme caractéristique locale (*local feature*) un motif d'un ou plusieurs pixels différents de son voisinage proche. Les différences observées s'appuient sur une ou plusieurs propriétés suivant les algorithmes : intensité des pixels, couleurs ou texture sur un ensemble de pixels. Les caractéristiques intéressantes sont typiquement des points, des arêtes ou un «patch» particulier (un ensemble de pixels carré voire rectangulaire). La recherche de caractéristiques locales peut s'effectuer directement sur une image en niveaux de gris ou sur une image binaire de contours obtenue à l'aide d'un premier traitement. Usuellement, on nomme «détection» la méthode permettant de déterminer et calculer les positions dans l'image des caractéristiques locales et «description» les mesures réalisées autour de ces caractéristiques locales. Généralement, la description est calculée selon une région centrée sur la caractéristique locale, de taille fixe (sur le même nombre de pixels) ou variable.

On distingue 3 grandes catégories d'usages faits à partir de caractéristiques locales :

— La première fait directement correspondre une information sémantique à la caractéristique extraite. C'est le cas par exemple en imagerie aérienne où une hypothèse associe toute courbe de fort contraste à une route.

— La deuxième n'associe pas de sens aux caractéristiques extraites mais se concentre sur la précision de leur localisation et représentation distincte dans l'image. Si leur détection est stable dans le temps, ces caractéristiques locales (nommé également *amers* dans ce contexte) permettent de calculer des poses relatives et de suivre des objets observés, procéder à la calibration de caméra ou encore d'engager une reconstruction 3D éparse de l'environnement. Le calcul de pose et les méthodes de suivi (*tracking*) permettent en outre d'aligner des images pour réaliser un panorama à l'aide d'image dont le champ de vue est proche (on parle de *mosaicing*). Un des premiers détecteurs dédié à cette tâche est le tracker KLT [LK+81].

— La dernière catégorie utilise un ensemble de caractéristiques locales comme la représentation robuste et compacte d'une image dans sa globalité. Ceci permet de reconnaître des classes d'objets (*object recognition*) ou encore d'associer des images représentant le même lieu (*visual place recognition*).

Invariance et covariance

Une méthode ou fonction est dite *invariante* à une certaine catégorie de transformations si sa réponse est identique quelque soit la transformation de cette catégorie appliquée à ses arguments. Une fonction est dite *covariante* si la transformation appliquée à ses arguments a le même effet qu'appliquer cette transformation à la réponse de

la fonction. Un cas courant en imagerie est celui de l'invariance aux rotations dans le plan de l'image : une fonction sera invariante si sa réponse est identique quelle que soit la rotation appliquée à l'image en entrée, mais sera covariante si sa réponse varie selon la même rotation que celle appliquée en entrée. Pour être invariante aux rotations, les méthodes d'extraction de caractéristiques réalisent souvent une normalisation via une rotation définissant l'axe principal de la caractéristique comme axe de référence pour la description établie.

Invariance aux rotations et caractéristique isotrope

Une fonction est isotropique en un point particulier si sa réponse est identique dans toutes les directions, caractère qui ne doit pas être confondu avec l'invariance aux rotations sus-mentionnée. On parle de caractéristique isotrope en particulier lorsque l'on s'attache à décrire une texture.

1.3.2 Propriétés d'une caractéristique locale idéale

On recense les propriétés suivantes nécessaires à des caractéristiques de bonne qualité [TM08] :

- *Répétabilité* : une caractéristique détectée dans une première image doit être détectée dans une autre image malgré des changements de conditions d'observation de la scène (éclairage ou point de vue par exemple)
- *Caractère discriminant* : les motifs décrits par les caractéristiques locales doivent être suffisamment variables et distincts afin d'associer au mieux les caractéristiques d'une image à l'autre
- *Localité* : les caractéristiques doivent être le plus «local» possible, c'est-à-dire représenter un ensemble de pixels restreint afin de réduire les risques d'occlusion et faciliter les problématiques d'estimation de transformations géométriques entre deux images
- *Quantité* : un compromis concernant le nombre de caractéristiques extraites doit être établi. En effet, plus le nombre de caractéristiques extraites est grand, plus on a de chance de décrire un objet de petite taille dans l'image. Mais un grand nombre de caractéristiques peut également conduire à des redondances dans l'information extraite et une représentation de l'image dans sa globalité non optimale. L'application visée détermine la densité de caractéristiques à extraire. En outre, la densité de caractéristiques extraites n'est pas constante dans l'image : les zones de fort contraste auront une réponse plus forte à la détection alors que les zones faiblement texturées retourneront peu ou pas de points (typiquement un ciel dégagé pour une scène extérieure). Il peut ainsi être nécessaire de forcer

une répartition équitable des points détectés dans toutes les zones de l'image pour en tirer une représentation à la fois compacte et complète.
— *Précision* : un point doit être détecté avec une localisation précise pour une utilisation ultérieure (en particulier pour les tâches de calibration et reconstruction 3D de l'environnement). Il doit en être de même pour sa taille et forme lorsque le détecteur prend en compte ces propriétés.
— *Performance* : Pour bon nombre d'applications, le temps de calcul nécessaire à l'extraction des points d'intérêt de l'image est une propriété déterminante dans les choix techniques effectués.

1.3.3 Détection basée sur les valeurs d'intensité de l'image

Historiquement, il s'agit de calculer la réponse d'une fonction C dite *corner response function* associant à chaque pixel de l'image un score de «saillance ponctuelle» (*cornerness*). Moravec dans [Mor80] définit cette fonction C comme la somme des différences au carré (SSD) entre un patch autour du point testé et des patches successivement décalés selon les coordonnées x y de l'image. Harris propose dans [HS88] un détecteur inspiré de ce concept en réalisant une approximation de la méthode de Moravec ; en utilisant la dérivée seconde de la SSD par rapport au décalage, il définit une fonction H comme suit :

$$H = \begin{bmatrix} \widehat{I_x^2} & \widehat{I_x I_y} \\ \widehat{I_x I_y} & \widehat{I_y^2} \end{bmatrix} \tag{1.9}$$

avec I_x et I_y respectivement les dérivées secondes en fonction de x et y, l'opérateur «chapeau» désignant la moyenne sur les patchs évalués. La fonction C est alors définie ainsi :

$$C = |H| - k(\text{trace}(H))^2 \tag{1.10}$$

Le détecteur proposé par Shi et Tomasi [ST94] fait appel à la même matrice H et effectue le calcul de ses valeurs propres λ_1 et λ_2, méthode présentée comme plus efficace face aux déformations affines. Ainsi, la fonction C devient la suivante :

$$C = \min(\lambda_1, \lambda_2) \tag{1.11}$$

D'autres contributions proposent des travaux dérivés relativement proches, faisant appel à la même matrice H en lui appliquant des métriques différentes. Zheng dans [ZWT99] propose également une modification de H pour calculer seulement deux images (dites «*smoothed*»).

Lowe introduit dans [Low04] l'idée d'un détecteur invariant aux changements d'échelle. Celui-ci est obtenu en convoluant l'image avec une gaussienne 2D à plusieurs échelles. Les différences entre les strates de la pyramide d'échelles obtenue (Difference of Gaussians or *DoG*) permettent de déterminer les points saillants et l'échelle à laquelle ils sont observables. La méthode *DoG* est en fait une bonne approximation de Laplacian of Gaussian (*LoG*), c'est-à-dire les dérivées secondes partielles sur les réponses aux convolutions gaussiennes. l'avantage de cette approximation est que le calcul est plus rapide. L'article [SB13] détaille une méthode quant aux choix des gaussiennes à appliquer pour obtenir une pyramide d'échelles optimale.

1.3.4 Méthodes de description

Une fois que nous avons nos régions issues de l'échantillonnage fixe ou entourant les points d'intérêt détectés, il faut les décrire, c'est-à-dire d'une certaine façon extraire de l'information de ces zones sous une forme plus compacte et plus facile à traiter.

Descripteurs globaux

Les descripteurs globaux se calculent la plupart du temps sur des images dans leur globalité [CRF11, NSP13, NSBS14], et quelques fois sur des patches de taille moyenne [LBKS13]. On calcule ainsi des critères entre deux images superposées tels que la «similarité propre» (*Self-similarity*) ou l'information mutuelle. Ces opérations permettent d'obtenir un score de vraisemblance entre deux images (ou deux patches d'images), et sont souvent utilisées au cœur de méthodes d'asservissement visuel [Dam10, DM12, CDM14].

Information mutuelle

D'autres méthodes ont été proposées dans l'état de l'art telle que GIST [OT01], les filtres de Gabor ou encore la représentation en ondelettes, qui sont une forme de représentation fréquentielles des informations spatiales de l'image.

Descripteurs de points d'intérêt

L'étape de détection permet donc de déterminer des coordonnées dans l'image qui correspondent à des points sources d'information (points saillants, forts contrastes, *etc*). Comme nous l'avons vu dans la partie précédente, ces détecteurs renvoient également pour certains des informations supplémentaires : une taille qui nous informe sur la région d'intérêt autour du point détecté et éventuellement une orientation (cela peut être la direction moyenne des gradients de la région par exemple). En fonction du détecteur choisi, on obtient en sortie de l'algorithme des patches de l'image qui sont de tailles variables et avec des orientations différentes.

Un descripteur est une opération mathématique qui nous permet de représenter ces points d'intérêt et leur région respective sous la forme d'un vecteur compact de scalaire ou d'une chaîne de bits (on parle aussi de decripteurs *binaires*). Avant de procéder à cette étape, il est nécessaire de réaliser une interpolation des régions à décrire lorsqu'elles ne sont pas de même dimension et orientation afin de les ramener dans des imagettes de résolutions identiques.

DESCRIPTEUR BINAIRE

Descripteurs scalaires

Ces techniques recueillent des informations de l'imagette à décrire sous la forme d'un ensemble de valeurs réelles (codées sur des nombres flottants ou entiers correspondant à la discrétisation d'une échelle choisie). Ces informations peuvent être des gradients ou une représentation fréquentielle spatiale par exemple. Ces vecteurs peuvent être comparés par la suite afin de déterminer s'ils représentent ou non le même point, par des distances courantes telles que les normes L^1, L^2 ou encore distances de Mahalanobis. Une comparaison des descripteurs scalaires est réalisée par [MS05]. On compte parmi ceux-là SIFT [Low04] qui reste très populaire pour son efficacité.

Le descripteur SIFT, dans sa proposition originale, considère une région de 16×16 pixels autour du point d'intérêt. Les gradients sont calculés sur cette imagette et pondérés par une gaussienne de manière à favoriser les gradients proches du centre de la région. La région est par la suite divisée en 16 sous-ensembles (4×4) et des histogrammes de gradients sont composés selon leur direction. Les histogrammes sont quantifiés selon 8 intervalles.

De nombreuses variantes et améloriations ont été proposées après SIFT. On peut citer par exemple SURF qui propose une approximation de la méthode employée par SIFT afin de réduire le temps de traitement au détriment de la précision. D'autre, comme affine SIFT, augmente la performance au détriment du temps de calcul afin de rendre le descripteur invariant aux transformations affines et envisage ainsi la reconnaissance de points d'intérêt dans deux images dont le point de vue a drastiquement changé.

Descripteurs binaires

Nous avons décrit rapidement dans la partie précédente la méthode de description employée dans la caractéristique SIFT. Nous pouvons en déduire que le descripteur obtenu est un vecteur de 128 scalaires pour chaque point représenté. Comparer deux points nécessite donc de calculer une métrique sur deux vecteurs de cette taille. Lorsque l'on cherche à établir la position d'un objet présent dans deux images, ou estimer une pose entre deux images à partir de points d'intérêts, cela nécessite d'extraire un grand

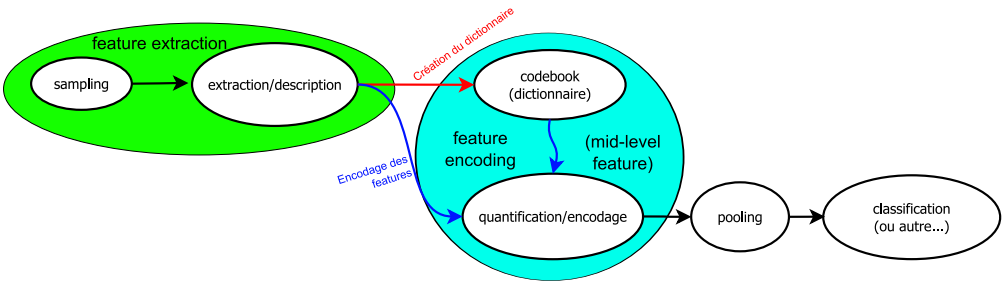


FIGURE 1.14 – Schéma global d'une approche avec calcul d'une représentation intermédaire

nombre de points pour ensuite les comparer deux à deux. Même si des méthodes, comme le RANSAC, ont été développées de façon à éviter de devoir tester toutes les combinaisons possibles, cela reste fastidieux et coûteux en temps de calcul.

Pour cette raison, de nombreux descripteurs binaires ont été proposés. Ceux-ci donnent en sortie une chaîne de bits (parfois nommé *hash* par analogie avec les condensats issus des fonctions de hachage) que l'on peut comparer grâce à la distance de Hamming. Les calculs d'appariement sont ainsi beaucoup plus rapides, même si intrinsèquement ces descripteurs encodent moins d'information et sont donc plus propices à une confusion entre deux points d'intérêt différents. Une comparaison effectuée sur plusieurs descripteurs binaires est compilée dans l'article [HDF12].

Compression de l'information : *Mid-level Features* et *pooling*

Le concept de *Mid-level Feature* (que l'on traduira ici par «représentation intermédiaire») a émergé dans la communauté de la «Recherche d'image par le contenu» (*Image retrieval*). Il s'agit de condenser l'ensemble des descripteurs correspondant à un objet ou une scène observée sous une forme plus compacte. La plupart du temps, les méthodes font appel à une méthode d'apprentissage non-supervisée sur un ensemble de descripteurs ponctuels pour composer un *codebook* ou dictionnaire. Chaque descripteur extrait sera quantifié par la suite sur ce dictionnaire. Une fois les descripteurs quantifiés, une méthode supplémentaire nommée *pooling* dans la littérature génère une représentation de l'image dans sa globalité ou de l'objet observé. La publication [CLVZ11] offre un aperçu et une comparaison de plusieurs techniques s'inscrivant dans le concept de *Mid-level features*. La figure 1.14 donne une représentation schématique d'une approche de description avec représentation intermédaire.

MID-LEVEL FEATURE

IMAGE RETRIEVAL

POOLING

La composition du dictionnaire peut se faire *via* différentes méthodes de *clustering* (*K-mean* ou mélanges de Gaussiennes par exemple). Quant à la quantification sur le

dictionnaire obtenu, la littérature distingue les cas de quantification «dure» (*hard quantization*) et «douce» ou «progressive» (*soft quantization*). On trouve dans la première catégorie la composition d'un histogramme de mots visuels (les étiquettes des clusters auxquels ont été associées les caractéristiques locales extraites des images) [PCI+08]. Dans la seconde, nous pouvons citer les méthodes *Kernel codebook encoding* [vGGVS08], *Local linear encoding* [WYY+10], *Fisher encoding* [PSM10] ou encore *Super-Vector encoding* [ZYZH10].

L'Approche Bag-of-Words

Bag-of-Words

L'approche *Bag-of-Words* ou «sacs de mots» peut être classée parmi ces techniques dites de représentation intermédaire. Cependant, son originalité mérite quelques mots supplémentaires. Il s'agit en effet d'une méthode qui a été inspirée par la recherche textuelle dans un large *corpus* de documents. Cette méthode est décrite plus en détails dans la publication [SZ03] : un dictionnaire est composé à l'aide d'un ensemble d'apprentissage et les descripteurs sont quantifiés sur ce dictionnaire. Il n'y a par contre pas de méthode de *pooling* si l'on s'en réfère à la définition donnée précédemment,

Inverted files

mais la création de «fichiers inversés» (*Inverted files*) qui à chaque mot du dictionnaire associent les images (par analogie avec les documents textuels) dans lesquels ils apparaissent ainsi qu'un coefficient *TF-IDF* (pour *Term Frequency-Inverse Document Frequency*). Ce coefficient permet de favoriser les mots visuels peu présents d'un document à l'autre et qui sont donc déterminants, de ceux que l'on retrouve dans la majorité des images et qui sont donc source de confusion.

Substitutions par réseaux de neurones à convolution (*CNN*)

Si l'on résume les étapes successives pour comparer deux images (requête et référence) à partir de points d'intérêt, on obtient la représentation proposée en figure 1.15.

Nous abordons ici des travaux liés à notre problématique, à savoir la tâche de requête basée images sur des images naturelles, en milieu extérieur et dans l'optique d'une reconnaissance de lieu. Certaines tâches du flux de traitement que nous avons rappelé précédemment peuvent être remplaceés par une méthode d'apprentissage (apprentissage profond par réseau de neurones à convolution surtout). C'est le cas de l'article [WKP16] qui remplace l'intégralité de la chaîne de traitement par une méthode CNN (figure 1.16).

D'autres travaux font appel aux réseaux de neurones pour des substitutions plus fines, notamment l'article [AAS+16] qui traite de la problèmatique de l'association de patches autour de points d'intérêt issus d'images de modalités différentes. On revient

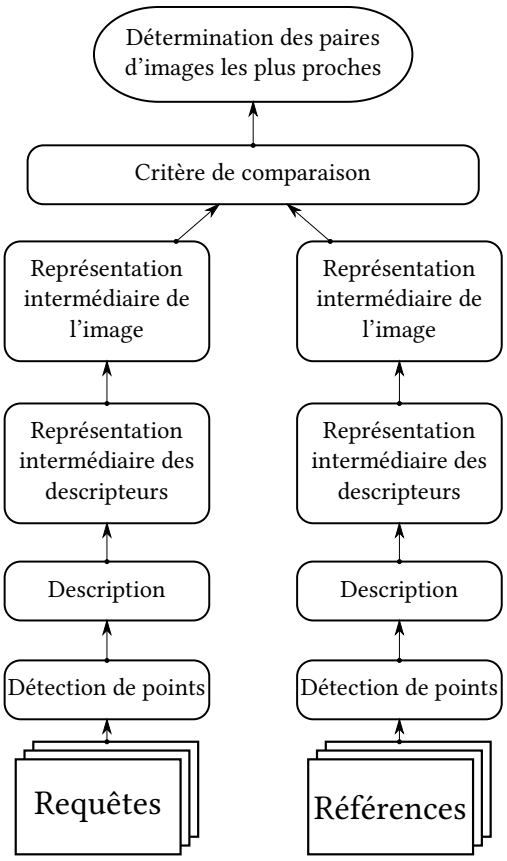


FIGURE 1.15 – Représentation graphique d'un flux de traitement pour comparer deux images à l'aide de points d'intérêt et d'une description intermédiaire.

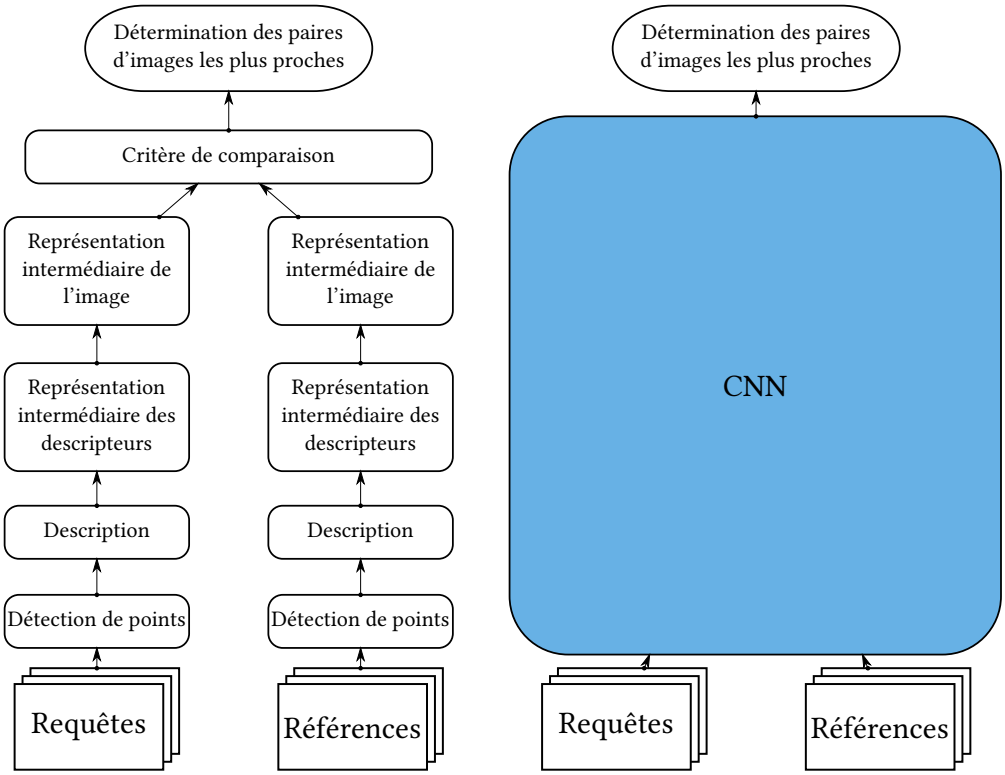


FIGURE 1.16 – Substitution de l'intégralité du traitement par un réseau de neurones convolutionnel.

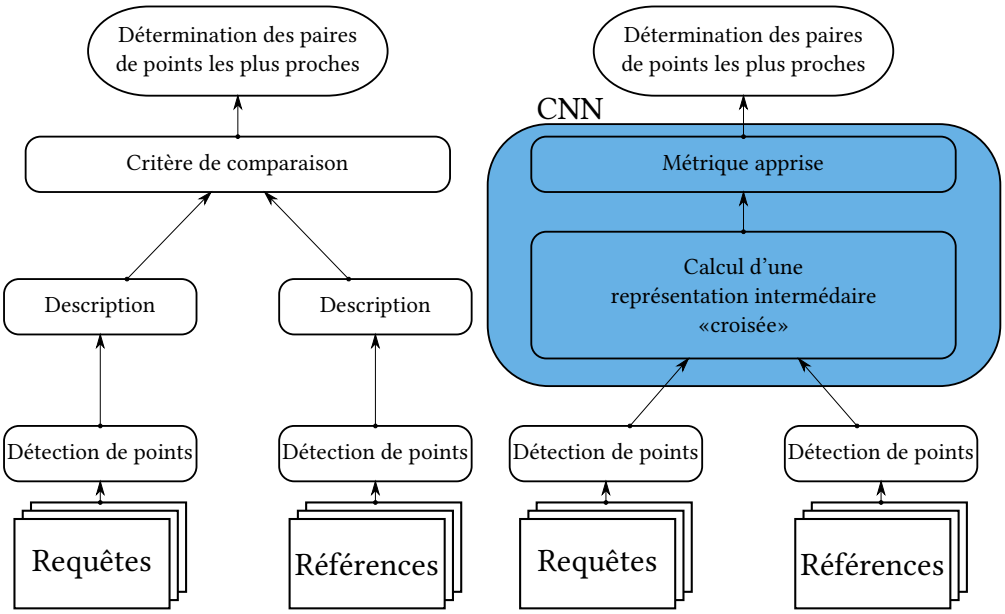


FIGURE 1.17 – Substitution des étapes de description et métrique de comparaison par un réseau de neurones convolutionnel.

ici à une chaîne de traitement plus courte où l'on se concentre sur des descriptions de points que l'on cherche à associer avec ceux d'une autre image (dans une optique d'association *baseline* courte) pour un calcul de pose relative par exemple, sans représentation intermédiaire de l'image comme pour une recherche dans une large base de données (figure 1.17).

Ces nouvelles méthodes émergentes proposent des résultats meilleurs que l'état de l'art. Néanmoins, la tâche d'apprentissage est très coûteuse en temps de calcul et efficace seulement avec un jeu de données d'apprentissage conséquent.

1.4 La problématique de la multimodalité

On parle de multimodalité lorsque l'on travaille avec des capteurs visuels aux caractéristiques différentes, particulièrement en terme de réponse spectrale. Alors que le sujet a été abordé dans le domaine de l'imagerie médicale, il est encore jeune dans le domaine de la vision pour la robotique.

Multimodalité et approches directes

Les approches directes font référence à l'usage des images dans leur globalité. la méthode décrite dans [CDM14] en est un exemple qui se focalise sur l'usage de l'information mutuelle pour effectuer des tâches de recalage d'images et de suivi (*tracking*). Dans [MB13], les auteurs utilisent une méthode de recalage d'images multimodales au cœur d'un processus de SLAM (*Simultaneous Localisation And Mapping* soit «Localisation et Cartographie Simultanées»). L'article [MV12] propose également une approche multimodale liant images visibles et infrarouges thermiques et annoncent des performances intéressantes de jour comme de nuit. Neanmoins, ces deux dernières références font usage des deux gammes spectrales au même moment, c'est-à-dire que les travaux menés utilisent toujours le même ensemble de capteurs joints quelques soient les expériences. Notre approche est différente : nous avons en effet fait le choix de nous placer dans l'hypothèse où deux systèmes différents peuvent faire usage de mêmes données. Ainsi, un système doit pouvoir faire une première acquisition et un deuxième, dans une autre modalité, doit être en mesure de pouvoir associer ses données avec celles du premier.

Des caractérisques ponctuelles adaptées à la multimodalité

Des travaux menés sur la question de la multimodalité en robotique, ou du moins sur l'association d'images de scènes naturelles, ont vu le jour ces dernières années. Certaines études sur le sujet ont été menées par [RCAC+14] par exemple. D'autres proposent des modifications à apporter à des extracteurs de caractéristiques locales pour les rendre invariantes au changement de modalité ont été proposées [FBS11, MA13].

Conclusion

Au cours de cet état de l'art, nous avons présenté brièvement le contexte dans lequel ces travaux de thèse ont été entrepris : celui de la robotique mobile et par extension le domaine de la navigation autonome pour le véhicule intelligent. Nous avons évoqué les capteurs pouvant être utilisés pour réaliser des tâches de localisation et mis l'accent sur les capteurs basés vision, sujet de cette thèse.

L'ensemble des capteurs basés vision, des méthodes de traitement image et des techniques de vision par ordinateur est très large. Pour une seule problématique, l'ensemble de la chaîne de traitement, du choix des caractéristiques physiques et électroniques du capteur aux solutions de traitements numériques appliqués par la suite nécessitent de réaliser des compromis face à des contraintes fortes.

L'observation d'une scène naturelle par un dispositif de vision passif est le fruit d'un processus complexe aux paramètres multiples : sources de lumière, type et orientation, nature des objets et leur interaction avec ces sources, caractéristiques physiques, optiques et géométriques du dispositif imageur, sensibilité spectrale du dispositif photosensible et sa réponse spectrale, *etc.* Pour un système évoluant en milieu naturel, il est impossible de connaître l'intégralité de ces paramètres si bien que, à défaut de pouvoir émettre des hypothèses crédibles sur les paramètres cachés et nécessaires au bon fonctionnement du système, on conçoit un système invariant, ou du moins robuste, aux variantions engendrées par la pluralité et la complexité des scènes possibles.

Les travaux de cette thèse se sont concentrés sur la problématiques de la localisation visuelle multimodale à long terme. La localisation visuelle est un exercice de recherche d'image par le contenu : en supposant que l'on possède une base d'images préétablies (voire une séquence) de scènes naturelles de l'environnement dans lequel notre système navigue, de nouvelles acquisitions d'images réalisées «en ligne» doivent être appariées avec les images de la base initiale afin d'en déduire que la scène observée est la même que celle enregistrée lors de la première expérience. Le système est ainsi soumis à deux grands ensembles de contraintes : les variations de l'environnement et la perception que le système en a à long terme d'une part, les changements de modalité lorsque les capteurs utilisés pour l'expérience initiale et la navigation en ligne sont différents d'autre part.

La perception que l'on a d'un environnement naturel au travers d'un système de vision est riche en variations au cours du temps. Les conditions d'ensoleillement, la direction de la lumière et les dégradations de perception engendrées par certaines conditions climatiques sont multiples. Le système est également soumis à d'autres acteurs dynamiques visibles dans les scènes étudiées (véhicules, piétions, *etc*) comme des éléments de l'environnement source de mouvement (végétation par exemple). À plus long terme, même les structures pouvant faire office d'amers rigides peuvent évoluer. En outre, les matières et textures des objets composant les images n'ont pas les mêmes comportements : les objets absorbent et réfléchissent différemment la lumière si bien que leur apparence peut être très différente d'une modalité à l'autre.

CHAPITRE

2

VISION MULTIMODALE VISIBLE/INFRAROUGE

Sommaire

Nous détaillons dans ce chapitre les contributions que nous avons apportées à la problématique de l'extraction de caractéristiques robustes à long terme et au changement de modalité. Dans une première partie nous détaillons les travaux menés sur un descripteur global. Dans une deuxième partie, nous nous intéressons au comportement des détecteurs de points courants face au problème de la multimodalité. Nous donnons des critères de choix ainsi que des pistes pour favoriser la répétabilité de ces méthodes d'une réponse spectrale à l'autre. Dans une troisième partie, nous proposons une nouvelle approche de description locale que nous avons baptisée PHROG. Nous présentons des résultats expérimentaux qui démontrent un apport en comparaison des méthodes de l'état de l'art.

2.1 Proposition d'un descripteur global

2.1.1 La mémoire : création d'une carte visuelle

Une carte visuelle est créée à partir de données provenant d'une première expérience réalisée avec un véhicule instrumenté. Ce véhicule dispose d'un GPS différentiel permettant d'obtenir une précision de la position mesurée de l'ordre du centimètre. Il est également équipé d'une caméra montée vers l'avant du véhicule. Le récepteur GPS nous permet d'associer précisément chaque image avec la position du véhicule au même moment. Nous appelons cette séquence vidéo la *mémoire*. Cette mémoire peut être comparée à une carte métrique avec des positions distinctives (ou *lieux*) pour lesquelles nous avons une vue acquise. Nous extrayons de chaque image de la mémoire une *signature d'image*, c'est-à-dire une caractéristique distinctive de l'image entière. La façon dont nous extrayons les signatures d'images pour la mémoire et la séquence en ligne est exactement la même. Nous allons par la suite comparer les signatures de la mémoire avec les signatures des images acquises en direct.

2.1.2 Calcul des signatures d'image

Méthode d'échantillonnage des caractéristiques

Nous avons choisi de décrire et de comparer des images avec des descripteurs globaux regroupant les descriptions de *patches* locaux. Nous réduisons la résolution des images et les divisons selon une grille régulière. La taille des imagettes obtenue est d'environ trente pixels pour la plupart des images. Si nous associons différents types de capteurs, nous devons être sûrs que les données sur les *patches* concernent approximativement les mêmes informations du monde physique. Contrairement aux méthodes de l'état de l'art qui définissent une grille arbitraire, nous proposons ici

d'utiliser une grille liée à la géométrie de l'optique de la caméra (ses paramètres intrinsèques).

Nous considérons le modèle du sténopé présenté dans la section 1.2.1 avec f_u et f_v les distances focales en termes de pixels pour les axes x et y, (u_0, v_0) les coordonnées du point principal en pixels, $X = [x, y, z, 1]^T$ les coordonnées homogènes d'un point 3D de l'environnement par rapport à la caméra et $\tilde{x}$ sa projection dans le repère des coordonnées de l'image. Nous définissons alors une sphère centrée sur le centre optique du modèle sténopé comme schématisé dans figure 2.1. Nous nommons α l'*angle d'ouverture*.

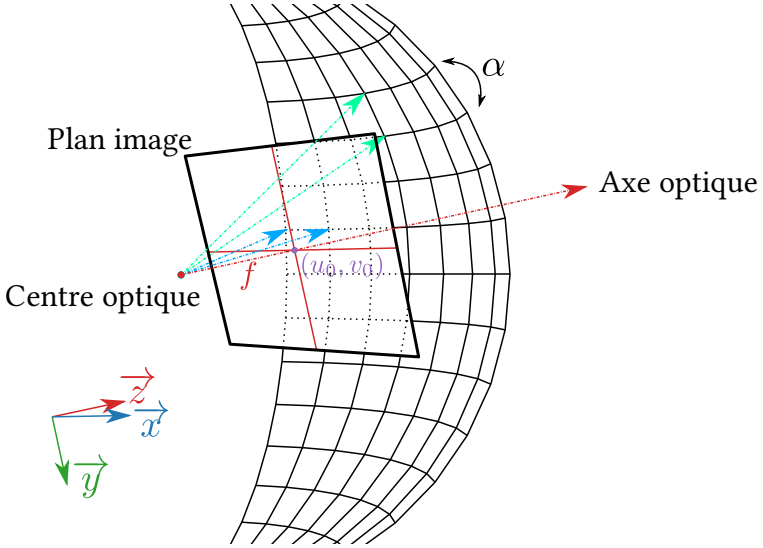


FIGURE 2.1 – Représentation schématique de l'échantillonnage de l'image basé sur des contraintes géométriques du capteur.

Les projections des directions espacées d'un angle α selon les axes $\overrightarrow{x}$ et $\overrightarrow{y}$ nous donnent les coordonnées liées aux *patches* selon la formule 2.1 (nous notons w_s la largeur du capteur photosensible en pixels, h_s sa hauteur). En effet, les coordonnées d'un point de l'espace X à l'intersection entre une sphère unitaire et une droite dans la direction donnée par les angles $m\alpha$ selon $\overrightarrow{x}$ et $n\alpha$ selon $\overrightarrow{y}$ peuvent s'exprimer sous la forme $X = [x, y, z, 1]^T = [-\sin(m\alpha), \sin(n\alpha), (\cos(m\alpha) + \cos(n\alpha)), 1]^T$.

$$\tilde{x}_{m,n} = \begin{pmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -\sin(m\alpha) \\ \sin(n\alpha) \\ (\cos(m\alpha) + \cos(n\alpha)) \end{pmatrix} \quad (2.1)$$

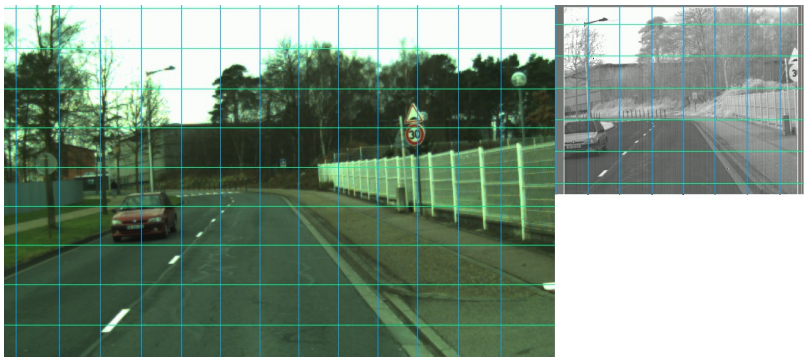


FIGURE 2.2 – Exemple de grilles d'échantillonnage calculées sur différents capteurs.

$$\text{pour tout } m \in \left[\!\left[\frac{\arctan\left(\frac{u_0 - w_s}{f_u}\right)}{\alpha} ; \frac{\arctan\left(\frac{u_0}{f_u}\right)}{\alpha} \right]\!\right]$$

$$\text{et tout } n \in \left[\!\left[\frac{\arctan\left(\frac{-v_0}{f_v}\right)}{\alpha} ; \frac{\arctan\left(\frac{h_s - v_0}{f_v}\right)}{\alpha} \right]\!\right]$$

Les valeurs calculées $\tilde{x}_{m,n}$ sont arrondies de sorte que les imagettes soient composées de pixels entiers. De cette manière, nous obtenons plusieurs portions d'image côte à côte. La taille des imagettes doit être choisie avec précaution : en effet, une grande taille réduirait les conséquences de variations faibles de point de vue (tel qu'expliqué par [NSBS14]), c'est pourquoi nous avons déterminé le meilleur angle α sur notre ensemble de données dans une gamme allant de 0° à 10° (voir section 2.1.4). Il est logique que les dimensions des imagettes, une fois projetées sur le plan du capteur, soient plus grandes à la périphérie de l'image qu'en son centre (un exemple est donné dans figure 2.2 avec $\alpha = 1°$). Comme nous pouvons le constater avec la figure 2.2, selon les caractéristiques de l'optique, nous n'avons pas nécessairement le même nombre de subdivisions dans nos images.

Histogrammes de gradients modifiés

Pour chaque subdivision, nous établissons des «Histogrammes de Gradients Orientés» (*HOG*). Nous appliquons ici la méthode décrite dans [DT05] : nous calculons dans un premier temps les angles des gradients pour chaque pixel et les agrégeons dans chaque carreau selon huit intervalles de directions. Nous obtenons ainsi des vecteurs composés de 8 scalaires. L'utilisation de ce descripteur permet une certaine robustesse face au changement de modalité [FBS11].

Nous ajoutons une étape à ce processus inspirée par [FBS11] et représentée dans la figure 2.4 : l'analyse empirique suggère aisément que certains objets ou matériaux apparaissent principalement noirs dans les spectres visibles alors qu'ils apparaissent dans des teintes claires dans les spectres infrarouges et *vice versa* (*cf* figure 2.3). Par conséquent, les orientations des gradients sont parfois inversées dans différentes plages spectrales. Pour rendre les histogrammes de gradient orientés invariants au sens des gradients, nous divisons les histogrammes représentés sur 360° en deux et sommons ensemble les gradients inclus dans les intervalles de sens opposés (figure 2.4).

FIGURE 2.3 – Une paire d'images infrarouge proche/visible provenant du jeu de données présenté dans la section 2.3.2. Des objets très contrastés (en particulier la végétation) dans le spectre visible (à droite), se confondent avec l'arrière-plan dans le spectre infrarouge (à gauche).

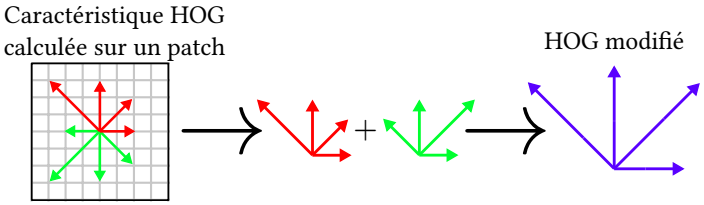


FIGURE 2.4 – Représentation schématique de la modification d'une description HOG de manière à la rendre invariante aux «inversions» du sens des gradients : les gradients de même directions mais de sens opposés sont additionnés. La taille des vecteurs de description est ainsi deux fois plus petite que celle des descriptions originales.

Concaténation des histogrammes

Les histogrammes de 4 valeurs résultants sont normalisés en fonction du nombre de pixels inclus dans la sous-région, puis stockés ensemble dans un tableau tridimensionnel pour obtenir une signature d'image.

2.1.3 Comparaison des signatures d'images

Nous désignons *séquence en ligne* une nouvelle expérience de navigation. Ce trajet peut avoir lieu plusieurs jours ou mois plus tard, de sorte que l'environnement présente des changements d'apparence importants. La seule entrée de l'algorithme considérée est le flux d'images provenant d'une autre caméra : l'optique et la taille du capteur peuvent changer, ainsi que la sensibilité spectrale. Nous calculons les signatures d'images avec le même paramètre d'angle d'ouverture dans notre méthode d'échantillonnage en grille et selon les nouveaux paramètres intrinsèques du capteur.

Afin de comparer deux images différentes, nous calculons la signature décrite précédemment pour chacune d'elles et utilisons la similartié cosinus pour calculer un score correspondant. La similarité cosinus est une mesure de similarité courante utilisée dans la recherche d'informations, en particulier dans la recherche textuelle [Sin01]. La similarité cosinus a été utilisée par [NSBS14] pour la reconnaissance visuelle de lieux et [MA13] pour la correspondance stéréo multimodale. Étant données deux signatures d'images σ_a et σ_b, le score de correspondance est donné par la formule suivante :

$$\text{score}(\sigma_a, \sigma_b) = \cos(\theta) = \frac{\sigma_a \cdot \sigma_b}{\|\sigma_a\|\|\sigma_b\|} \tag{2.2}$$

La similarité cosinus par définition est toujours définie dans l'intervalle $[0, 1]$. Dans le cas où nous avons des signatures d'images de différentes tailles, nous essayons toutes les possibilités de superposition de la plus petite grille sur la plus grande de gauche à droite et du haut du champ de vision vers le bas, calculons tous les scores de similarité possibles et conservons seulement le meilleur.

2.1.4 Résultats expérimentaux

Nous avons divisé nos expérimentations présentées dans cette partie selon deux axes : d'abord, nous avons recherché une valeur pour l'angle d'ouverture α optimale. Deuxièmement, nous avons entrepris des expériences pour vérifier si notre approche est assez distincte sur les données réelles et suffisamment invariante aux changements de gamme spectrale.

Nous avons acquis un premier ensemble de données afin de vérifier si l'approche globale proposée est elle-même suffisamment discriminante mais aussi robuste pour associer les données issues de caméras radicalement différentes. Cet ensemble de données condense un trajet à la fois dans des zones urbaines et une 4-voies et a été réalisé avec trois caméras différentes : deux caméras visibles identiques avec un écart (*baseline*) de 30 cm et une troisième entre les deux précédentes. Cette dernière est une caméra SWIR (*Short Wavelength Infrared*). Un échantillon d'images visibles et SWIR a été donné dans figure 2.2. Chaque séquence est composée de 200 images.

Méthode d'évaluation des algorithmes de recherche

Nous appliquons ici les critères d'évaluation usuels des méthodes de recherche de données. Prenons un espace de recherche (figure 2.5) dans lequel on souhaite sélectionner un ensemble d'élément pertinent (le cercle rouge sur le schéma). L'algorithme évalué pour cette tâche renvoie un ensemble d'éléments qui ne correspond pas forcément exactement à l'ensemble des éléments pertinents (le cercle bleu) :

- une partie des éléments pertinents n'est pas détectée par l'algorithme (zone α sur le schéma). On nomme ces éléments «faux négatifs» (*False Negatives*, *FN*) ;
- une partie des éléments pertinents est retournée par la méthode (zone β). On nomme ces éléments «vrais positifs» (*True Positives*, *TP*) ;
- un ensemble d'éléments supplémentaire non pertinents est renvoyé par l'algorithme (ensemble γ sur le schéma). On nomme ces éléments «faux positifs» (*False Positives*, *FP*).

On peut alors évaluer les critères de «Précision» (*Precision*) et «Rappel» (*Recall*) en fonction du nombre d'éléments inclus dans chaque ensemble ainsi défini (équations (2.3) et (2.4)).

$$\text{Précision} \triangleq \frac{TP}{TP + FP} \tag{2.3}$$

$$\text{Rappel} \triangleq \frac{TP}{TP + FN} \tag{2.4}$$

Nous allons pouvoir calculer ces deux rapports à l'aide d'une matrice de confusion (ou matrice de similarité). Une première matrice de similarité est donnée en figure 2.6. Celle-ci permet de représenter les valeurs des mesures de similarité établies entre les éléments d'un ensemble de recherche (selon une direction) et les éléments «requêtes», à associer à ceux du premier ensemble (selon l'autre direction). Les méthodes d'indexation utilisent en général un seuil à appliquer sur ces valeurs de façon à déterminer si deux éléments comparés appartiennent à la même classe. Une manière d'évaluer

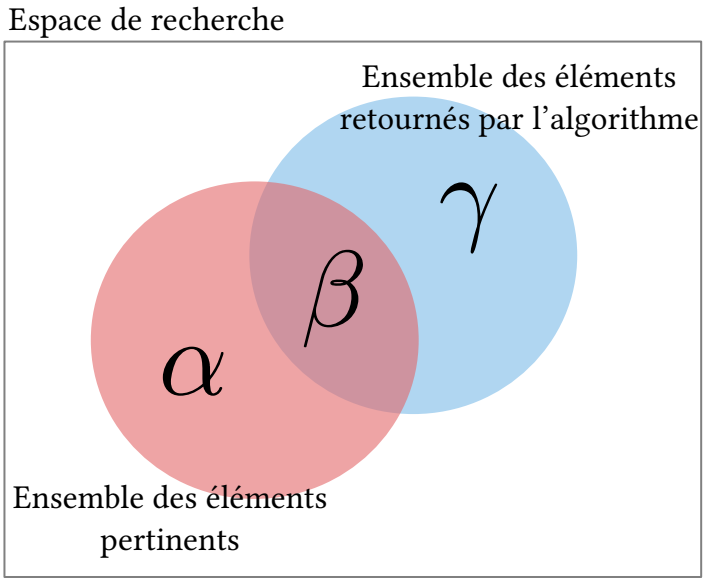


FIGURE 2.5 – Représentation graphique des résultats d'une méthode de recherche dans un espace. L'ensemble des éléments pertinents et des éléments retournés ne sont pas identiques.

une méthode d'indexation consiste à faire varier ce seuil de la valeur la plus haute présente dans la matrice de confusion, à la valeur la plus basse. À chaque valeur de seuil, on calcule alors les critères de précision et rappel. Ces couples de valeurs vont nous permettre de tracer une courbe «Précision-Recall» (figure 2.9 par exemple). Les courbes PR ont été largement utilisées pour évaluer les processus de recherche et indexation : un classifieur parfait devrait présenter une courbe PR allant des points de coordonnées $(0, 1)$ à $(1, 0)$ avec une AUC (*Area Under The Curve* ou aire sous la courbe) la plus proche possible de 1. Avec cette considération, la meilleure méthode d'un ensemble est celle qui présente l'AUC la plus élevée.

Nous avons synchronisé nos trois caméras avec un déclencheur matériel dédié (*hardware trigger*). Nous avons ensuite pris plusieurs séquences vidéo et calculé la matrice de similarité entre les séquences des deux caméras visibles. Cette expérience permet de vérifier si une petite variation du point de vue infère sur la mesure de similarité. Le calcul sur le premier enregistrement est donné dans la figure 2.6. Des scores plus élevés demeurent sur la diagonale de la matrice de similarité, la caractéristique choisie est suffisamment discriminante et engendre très peu de fausses correspondances. Les courbes de précision-rappel pour une correspondance visible à visible en fonction de la valeur α sont données dans la figure 2.7.

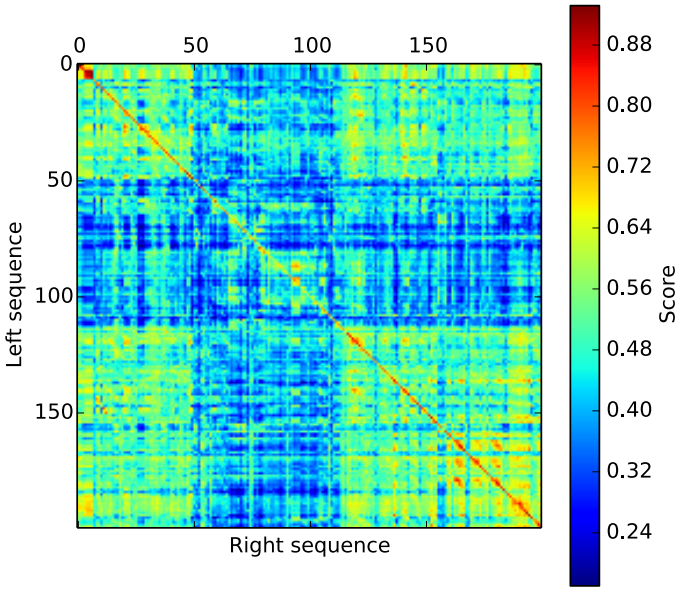


FIGURE 2.6 – Matrice de similarité calculée sur deux séquences visibles synchronisées.

Nous avons appliqué le même processus avec une paire de caméras visible et SWIR (figure 2.8 et figure 2.9). Cette tâche est beaucoup plus difficile que dans le précédent cas, mais la mesure de similarité propose néanmoins des résultats convenables. Par la suite, nous essayons également sur cet ensemble de données plusieurs valeurs pour l'angle d'ouverture α déterminant la taille des grilles générées dans les images. Les valeurs testées vont de 1° à 10° pour l'angle d'ouverture. Nous avons ajouté dans la figure 2.9 une comparaison avec une méthode plus commune associant caractéristiques locales SIFT et approche Bag-of-Words avec un dictionnaire de 1000 mots visuels.

Un découpage défini par un angle d'ouverture de 2° se révèle être le meilleur compromis pour l'association visible/visible et visible/SWIR. Avec de tels paramètres, le calcul de l'association entre les deux séquences dure 1 minutes et 30 secondes sur une machine de bureau équipée d'un processeur *Intel core i5* et $8Gio$ de RAM.

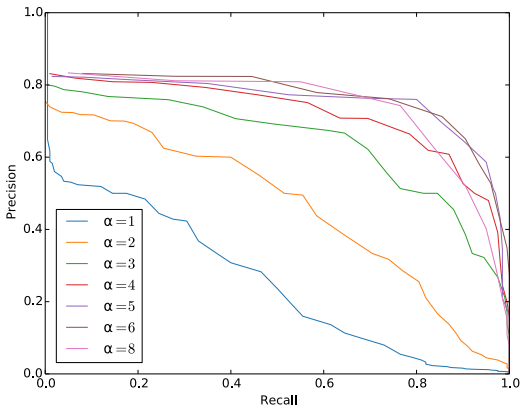


FIGURE 2.7 – Courbe Précision-Rappel de l'association d'images visibles et visibles.

2.1.5 Discussion

Nous avons développé dans cette partie un descripteur global pour la localisation visuelle. Cette approche utilise des paramètres géométriques donnés par une matrice des paramètres de calibration courants afin de comparer les données fournies par différentes caméras (optique, taille du capteur). La particularité de ce travail repose sur son approche multi-capteurs. La contribution principale ici consiste à utiliser deux caméras différentes pour la tâche de cartographie d'une part et la tâche de localisation d'autre part. De plus, nous utilisons des caméras ayant une sensibilité sur une bande spectrale très différente : spectre visible et SWIR. De tels choix techniques visent à assurer l'interopérabilité entre des capteurs très différents en vue de futurs systèmes partageant la même carte visuelle.

2.2 Analyse de détecteurs de points courants face à la multimodalité

Historiquement, les premiers détecteurs de caractéristiques locales étaient des détecteurs de coins : ils utilisaient le voisinage proche de chaque pixel pour distinguer les coins et les pixels non pertinents. Les références [HS88, ST94] calculent la carte de *cornerness* d'une image grâce au calcul des matrices d'auto-corrélation et du critère sur leurs valeurs propres comme nous l'avons vu dans le chapitre consacré à l'état de l'art. Plus tard, des études ont été menées sur des détecteurs avec la capacité d'estimer une caractéristique et son «échelle» appropriée pour la description afin d'améliorer la

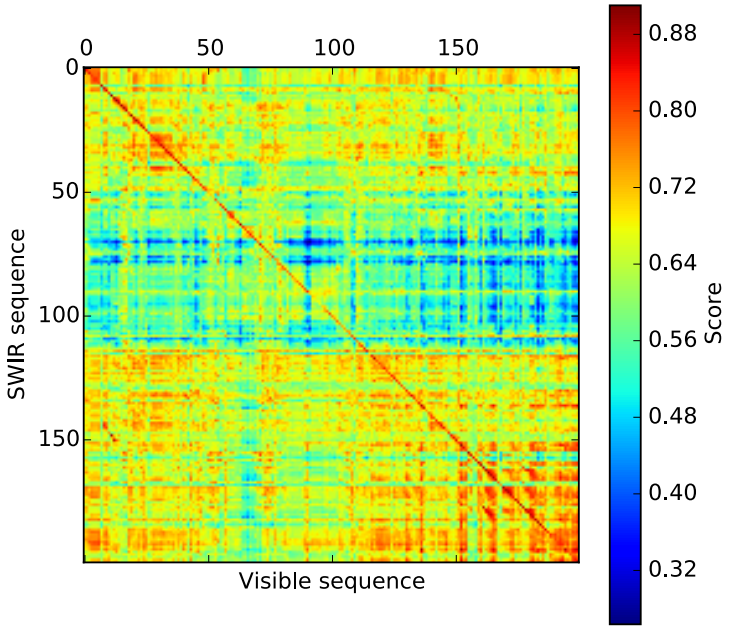


FIGURE 2.8 – Matrice de similarité calculée sur deux séquences, visible et SWIR, synchronisées.

reconnaissance des caractéristiques même si le point de vue a changé. Il en résulte des méthodes telles que SIFT [Low04] ou SURF [BETVG08] qui utilisent respectivement les «Differences de Gaussiennes» ou une formule approximative pour trouver une caractéristique et son échelle. Les caractéristiques sont alors plus proches des «*patches* d'intérêt» plutôt que des coins. Ces ensembles de pixels connectés sont généralement appelés *blobs*.

BLOBS

Comme l'auteur le remarque dans [AAS+16], les objets apparaissent différemment dans les images infrarouges et les images couleur. Les objets perdent globalement leurs textures : par exemple, une sérigraphie avec de nombreuses couleurs et formes semble homogène dans les images infrarouges (Figure 2.10). Un autre exemple remarquable est celui de la végétation qui apparaît beaucoup plus sombre dans les images visibles que dans le spectre infrarouge (un paysage semble «neigeux» comme dans Figure 2.11).

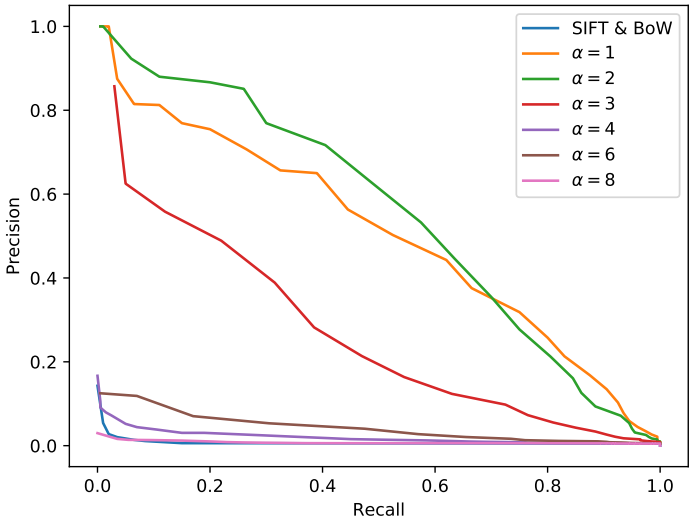


FIGURE 2.9 – Courbe Précision-Rappel de l'association d'images visibles et SWIR.

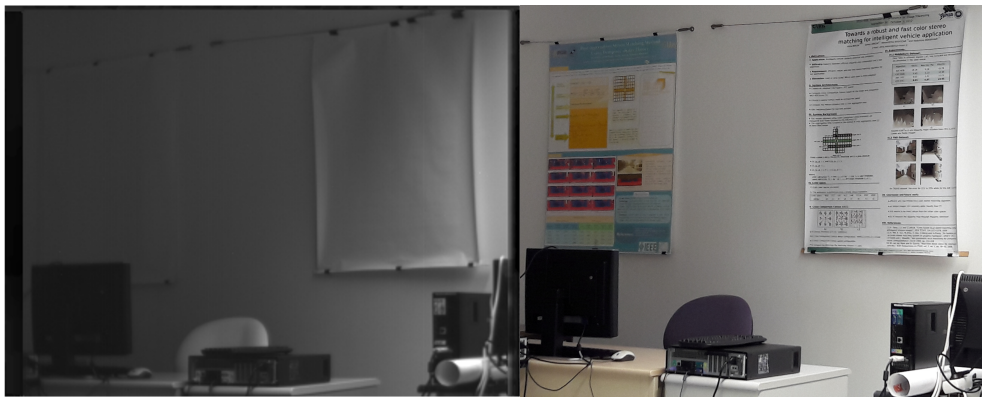


FIGURE 2.10 – Exemple de sérigraphie observée dans le spectre visible et le spectre infrarouge : alors que les formes imprimées sont perceptibles dans le visible, le poster semble vierge dans le spectre infrarouge.

Nous avons ainsi entrepris de tester quelques détecteurs de points disponibles dans la bibliothèque OpenCV [1] afin d'oberver leur réponse sur des images de la même scène à travers des modalités différentes et avec les mêmes paramètres.

1. http ://opencv.org/downloads.html

FIGURE 2.11 – Une paire infrarouge-visible extraite de notre jeu de données et introduite dans section 2.3.2.

2.2.1 Observations qualitatives

Le détecteur de Shi-Tomasi par exemple, avec ses paramètres par défaut, renvoie une réponse très différente d'une modalité à l'autre (figure 2.12). En jouant sur les paramètres (nombre de points maximum et qualité), on obtient des réponses plus proches entre les deux modalités, en particulier pour les points détectés au niveau du trottoir (figure 2.13).

Il en est de même pour le détecteur de Harris avec les paramètres par défauts (figure 2.14), mais en abaissant le paramètre de qualité des points retournés, les réponses sont plus semblables (figure 2.15).

Le détecteur FAST avec les paramètres par défaut renvoie des points visiblement bien répartis dans l'image et qui semblent conformes d'une modalité à l'autre (figure 2.16). À noter que la méthode de détection nécessite un motif de plus grande taille que pour les détecteurs de Harris et Shi-Tomasi.

Le détecteur de SIFT est plus complexe du fait qu'il évalue une taille de la zone d'intérêt ainsi qu'une orientation (on parle d'ailleurs parfois plutôt d'un détecteur de *blobs* ou *patches*). Celui-ci est donc plus sensible aux changements de textures et on retrouve peu de caractéristiques détectées à la fois dans une modalité et l'autre (figure 2.17). Jouer sur les paramètres de seuil permet de détecter plus de points communs aux deux modalités, mais provoque la détection de points parasites de petite taille dus au bruit des images considérées (figure 2.18).

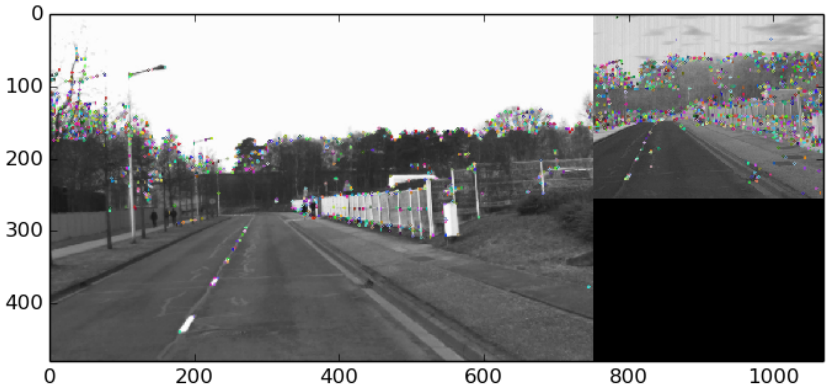


FIGURE 2.12 – Réponses du détecteur de Shi-Tomasi avec les paramètres par défaut.

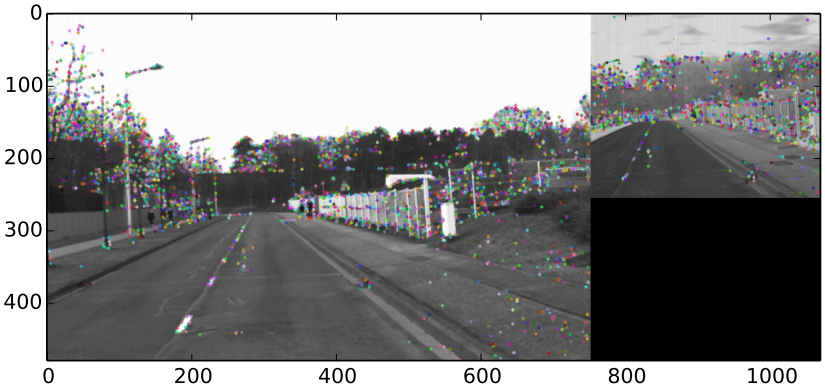


FIGURE 2.13 – Réponses du détecteur de Shi-Tomasi après ajustements manuels.

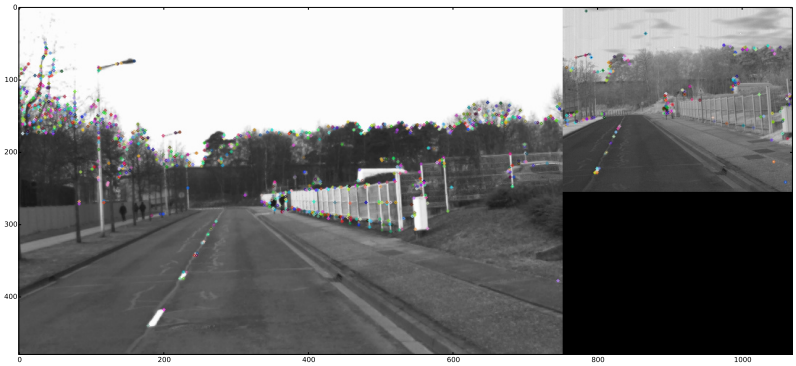


FIGURE 2.14 – Réponses du détecteur de Harris avec les paramètres par défaut.

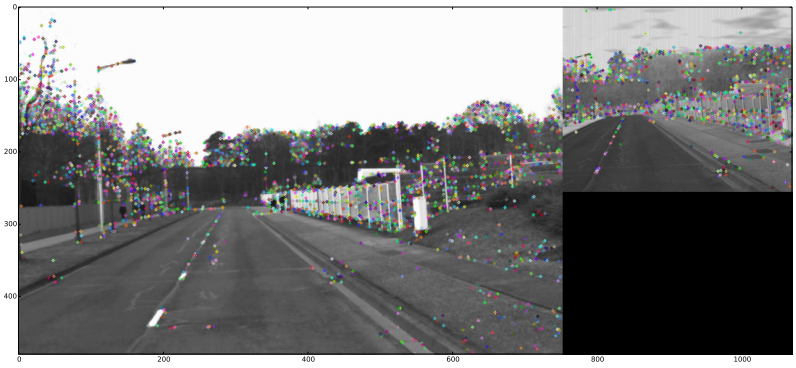


FIGURE 2.15 – Réponses du détecteur de Harris après ajustements manuels.

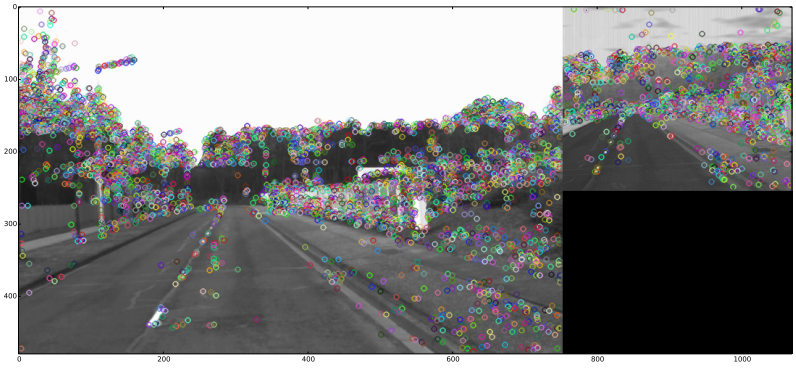


FIGURE 2.16 – Réponses du détecteur FAST avec les paramètres par défaut.

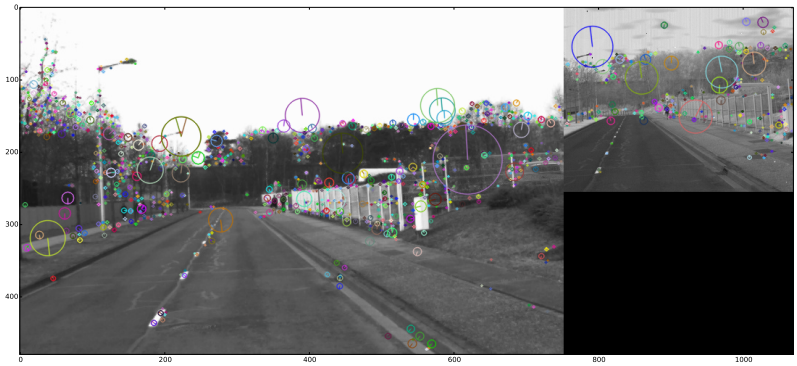


FIGURE 2.17 – Réponses du détecteur SIFT avec les paramètres par défaut.

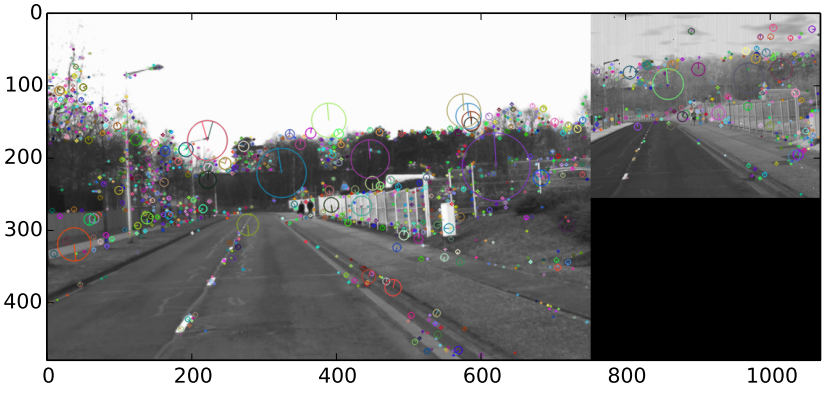


FIGURE 2.18 – Réponses du détecteur SIFT après ajustements manuels.

2.2.2 Critères de choix des paramètres pour envisager un réglage automatique

Voici plusieurs critères envisagés pour qualifier la réponse d'un détecteur sur une image :

- le ratio nombre de points détectés/nombre de pixels dans l'image : dans les exemples précédents, celui-ci oscille entre 0,70% et 1%. Le problème de ce critère est qu'il est global, et on peut avoir une concentration de points dans une zone particulière de l'image, avec des caractéristiques qui se superposent et donc correspondent à une même source d'information ;
- évaluer la densité des points détectés dans l'image : calculer le même ratio que précédemment mais par zones de l'image (en considérant des zones distinctes ou par fenêtre glissante). On peut ainsi éventuellement «forcer» la détection dans les zones pauvres en points ;
- Prendre en compte l'entropie de la zone considérée : typiquement une zone homogène (de faible entropie) présente peu d'information utile. Néanmoins, des zones très texturées ont une forte entropie alors que les textures sont réputées pour être difficilement associables d'une modalité à l'autre.

2.2.3 Tests préliminaires sur la répétabilité des détecteurs

Sachant ce qui se passe globalement avec les textures dans différentes plages spectrales, nous avons décidé de valider l'étape de détection de caractéristiques elle-même afin de choisir la meilleure approche de détection, à savoir la détection d'angles ou la détection de *patches* d'intérêts. Nous avons pris des paires d'images à partir de l'ensemble de données LWIR visible introduit dans [AST15]. Chaque paire a été ajustée afin de rendre les points de vue et les résolutions des deux images (infrarouges et visibles) identiques. Nous avons ensuite exécuté des algorithmes de détection pour chaque modalité et vérifié si les caractéristiques se trouvent dans les deux modalités. Nous avons concentré nos tests sur le détecteur de coin Harris ainsi que le détecteur «Differences de Gaussiennes» tel qu'utilisé dans la bibliothèque SIFT OpenCV. Nous avons considéré la répétabilité comme critère d'évaluation, qui est donné, pour chaque paire, comme un rapport entre le nombre de points d'intérêt détectés à la même position dans les deux images, au nombre total de points d'intérêt renvoyés par l'algorithme dans les deux images. Les paramètres de chaque algorithme ont été réglés pour obtenir les meilleurs résultats selon la méthode suivante : nous avons calculé la répétabilité sur une séquence complète avec plusieurs valeurs pour chaque paramètre de détecteur. Les figures 2.19 et 2.20 montrent les résultats obtenus avec le détecteur Harris lors de l'ajustement respectivement des paramètres de *niveau de*

qualité et *distance minimale* entre deux points détectés contigus. Le réglage du ratio de qualité peut apporter des améliorations significatives à la répétabilité de certaines paires d'images. Globalement, les résultats sont meilleurs lorsqu'un niveau de qualité faible (égal à $0,0001$) est choisi. La distance minimale entre deux pixels sélectionnés comme caractéristiques est un critère moins déterminant sur la répétabilité et nous avons choisi de garder une distance minimale de 2 pixels afin de détecter moins de points tout en préservant la répétabilité.

Nous définissons une tolérance de deux pixels dans les positions renvoyées par chaque algorithme. La figure 2.21 donne les résultats de répétabilité en fonction de chaque paire de données visibles-LWIR [AAS+16]. Pour chaque paire d'images, la répétabilité est bien meilleure avec une approche de détection similaire à celle du coin. Nous supposons que ces résultats sont dus au fait que les formes de l'objet sont essentiellement les mêmes dans les deux modalités, alors que les textures ont tendance à différer. La conception des caractéristiques en coin est plus corrélée aux formes de l'objet que les détecteurs de type *patches*. Par conséquent, nous avons choisi d'utiliser la méthode Harris comme procédure de détection.

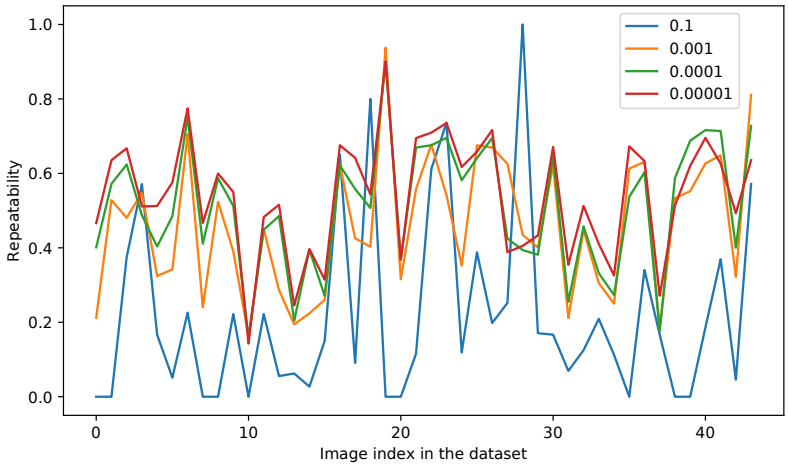


FIGURE 2.19 – Ajustement des paramètres du détecteur : répétabilité des points détectés en fonction des images de la séquence et du paramètre de qualité.

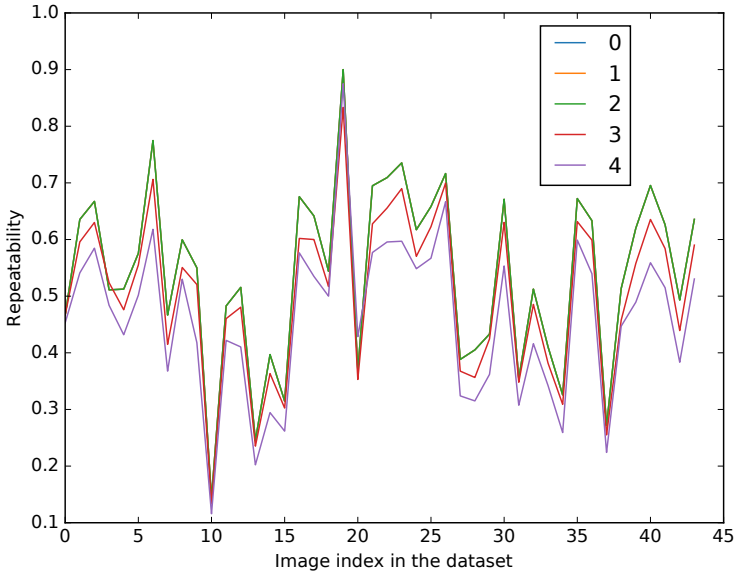


FIGURE 2.20 – Ajustement des paramètres du détecteur : répétabilité des points détectés en fonction des images de la séquence et de la distance minimale (en pixels) entre deux points détectés.

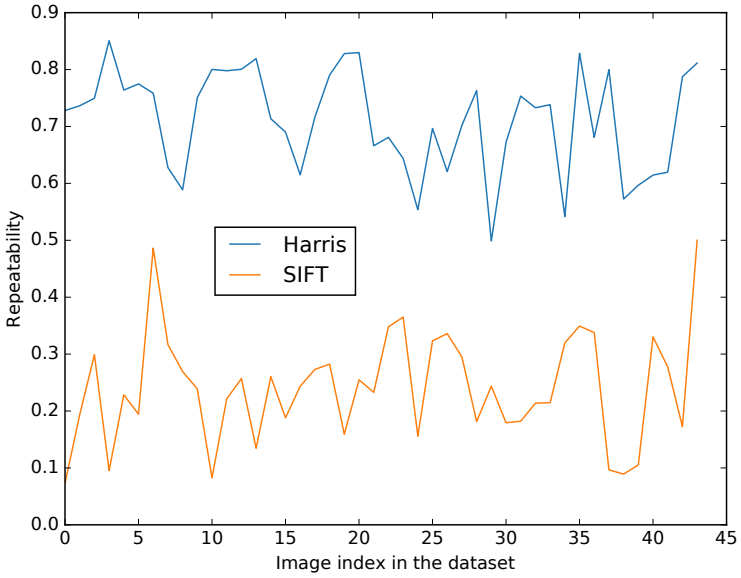


FIGURE 2.21 – Répétabilité des détecteurs Harris et SIFT pour chaque paire d'images du jeu de données visible-infrarouge lointain (LWIR).

2.3 Proposition d'un descripteur ponctuel : PHROG

2.3.1 Méthodologie

Une des méthodes les plus courantes dans la communauté de la robotique permettant d'estimer la pose d'une caméra est la recherche par le calcul de la *matrice fondamentale* [HZ03]. À l'aide d'une approche de type *RANSAC* (*RANdom SAmple Consensus*), ce calcul permet d'éliminer les faux appariements. La figure 2.22 est un exemple d'appariements restant après extraction de points SIFT et calcul de la matrice fondamentale entre une image visible et une image infrarouge. La différence d'apparence entre les deux modalités spectrales est telle que les descriptions de mêmes points de l'espace sont peu semblables, si bien que l'algorithme ne converge pas vers une solution satisfaisante. Cette constatation nous a amené à nous tourner plutôt vers des approches développées dans le domaine de la *recherche d'image par le contenu* (*content-based image retrieval*).

RANSAC

IMAGE RETRIEVAL

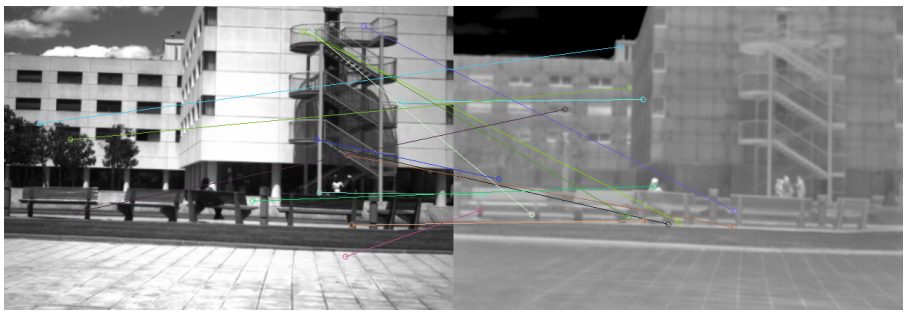


FIGURE 2.22 – Calcul de la matrice fondamentale à l'aide de points SIFT extraits. Les nombreux faux appariements ne permettent pas à l'algorithme de converger vers un résultat cohérent.

Extraction des caractéristiques : détection et description

Une façon courante d'extraire les informations pertinentes des images est de choisir un type de caractéristique particulière : les caractéristiques peuvent être des points, des régions, des arêtes ou des lignes droites par exemple. Les points d'intérêt locaux sont les plus courants dans les applications SLAM ou *Structure-From-Motion*, principalement parce qu'ils permettent un calcul plus approfondi des relations géométriques entre plusieurs images ou une reconstruction 3D éparse du milieu environnant. De nombreuses méthodes d'extracteurs de caractéristiques différentes ont été proposées dans la littérature. Le processus de calcul est habituellement divisé en deux parties :

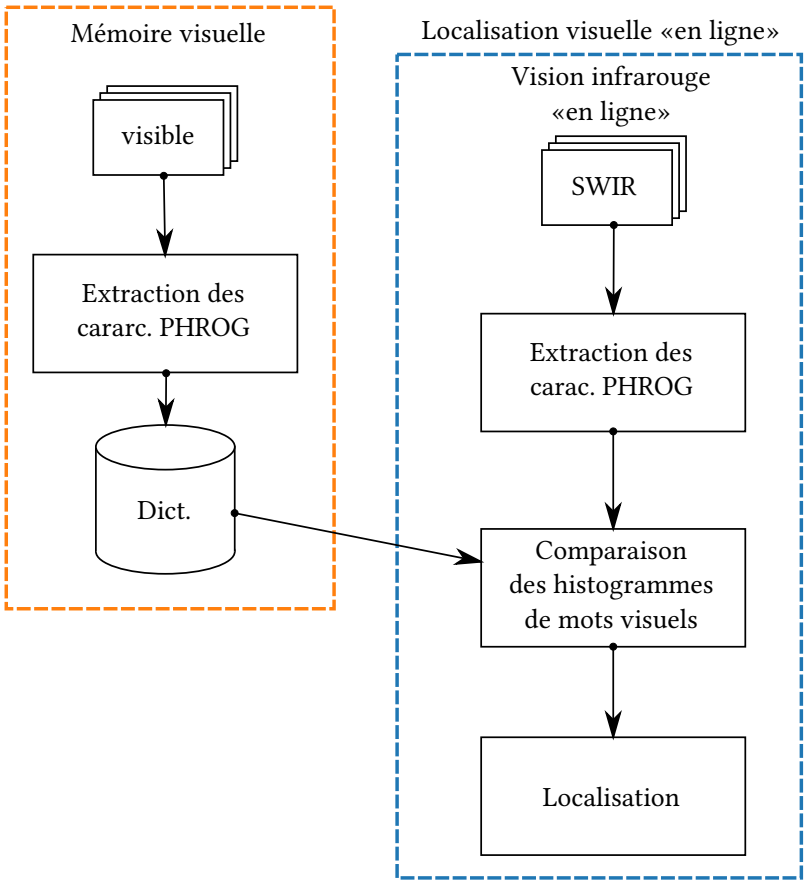


FIGURE 2.23 – Vue globale de la méthode proposée : les caractéristiques PHROG sont extraites de chaque image d'une séquence que l'on considère comme la mémoire. Un dictionnaire est déterminé par les centres des clusters calculés. Chaque image issue d'une nouvelle séquence que l'on nomme «en ligne» est comparée par la suite successivement avec chaque image de la mémoire en fonction de leurs histogrammes de mots visuels.

la détection et la description du point d'intérêt. Le lecteur peut se référer à deux comparatifs sur les détecteurs [MTS+05, TM08], une étude sur les descripteurs locaux [MS05] et une étude comparative sur les descripteurs binaires [HDF12] pour obtenir des détails pratiques sur les différents algorithmes impliqués. Les descripteurs binaires ont été favorisés depuis quelques années pour leur vitesse de calcul, mais font face au problème d'«inversion de gradient» posé par la multimodalité comme expliqué dans la section 2.3.1. Nous proposons dans cette partie une nouvelle approche de description que nous nommons PHROG.

Motif de description multi-échelle

Compte tenu de la nature du détecteur de Harris, nous n'avons aucune information préalable concernant l'échelle du *patch* de l'image à décrire. Par conséquent, nous nous sommes résolus à établir plusieurs descriptions à différents niveaux d'échelle. De manière analogue à ce qu'il se fait pour des caractéristiques invariantes à l'échelle comme *SIFT*, nous calculons une «Pyramide de Gaussiennes» à cette fin, le premier niveau d'échelle étant l'image originale. Pour composer les niveaux suivants, nous convoluons l'image source avec un noyau Gaussien de 5×5 pixels pour lisser et enlever ses composantes à hautes fréquences. Nous sélectionnons ensuite l'image résultante en prenant un pixel sur deux selon les deux axes x et y. Nous faisons de nouvelles itérations comme celle-ci en fonction du nombre de niveaux d'échelle désiré. Au cours de nos expérimentations, comme nous avons obtenu des images à partir de capteurs ayant des résolutions différentes et avec des points de vue variés, nous avons évalué l'efficacité de PHROG avec plusieurs niveaux d'échelle de description différents. Nous utilisons les courbes Precision-Recall et leur AUC (*Area Under the Curve* pour «aire sous la courbe») comme une évaluation de la performance de PHROG avec différents paramètres d'échelle. Nous donnons, dans la figure 2.24, un exemple des résultats obtenus sur l'ensemble de données VPrice (qui est présenté dans Section 2.3.2). Nous constatons que 5 niveaux de description sont appropriés car le calcul des niveaux supplémentaires n'entraîne aucun avantage significatif sur les résultats correspondants et constitue un bon compromis pour le coût mémoire de notre proposition. Nous ne faisons aucun autre raffinement d'angle ou interpolation d'image supplémentaire.

Histogrammes de gradients orientés réduits

Sur chaque niveau, nous composons un descripteur inspiré par le motif utilisé dans SIFT (voir figure 2.25) : nous considérons un voisinage de 4×4 zones de 4×4 pixels. Les zones centrales se chevauchent de sorte que le pixel d'angle (la position centrale de la caractéristique) est inclus dans 4 zones et chaque pixel restant à la même abscisse ou ordonnée appartient à 2 zones. Contrairement au descripteur SIFT, nous n'effectuons aucune pondération supplémentaire sur la valeur d'intensité des pixels avant le traitement suivant : compte tenu de la zone de chevauchement, les informations des pixels centraux sont déjà considérées deux fois par rapport aux autres pixels (et même quatre fois pour le pixel central).

Pour chaque zone donnée, nous calculons un histogramme de gradients orientés de taille N. Le problème des «gradients inversés» dû à l'imagerie multispectrale est expliqué dans [FBS11] : un matériau particulier a des propriétés de réflectance différentes selon les plages spectrales, de sorte que deux matériaux différents peuvent

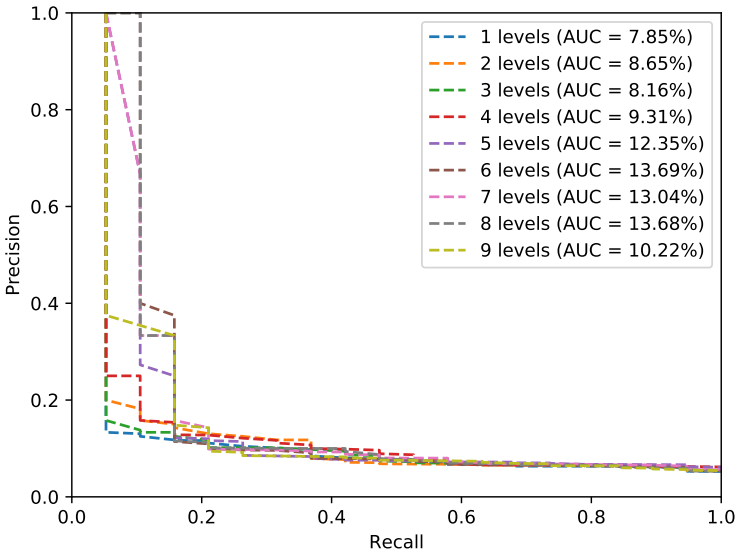


FIGURE 2.24 – Courbes Precision-Recall et leur AUC relatives (aire sous la courbe) en fonction du nombre de niveaux de description utilisés dans PHROG, appliqué sur le jeu de données VPRiCE.

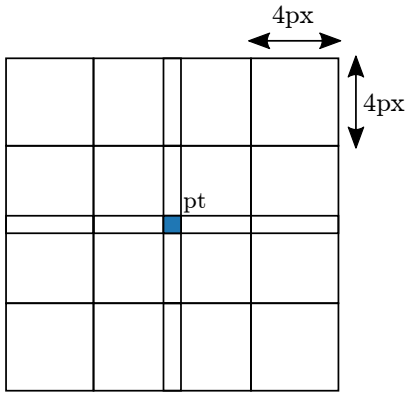


FIGURE 2.25 – Motif utilisé pour la phase de description. Celui-ci définit les zones à extraire autour du point d'intérêt.

avoir des réponses variables lorsqu'une scène est observée avec plusieurs capteurs. En particulier, une zone de contraste élevé dans une image au niveau des bords d'un objet composé d'un matériau donné devant un autre objet avec un matériau différent peut apparaître comme son propre négatif avec un autre capteur : les pixels blancs dans la première image apparaissent comme noirs dans la seconde et *vice versa*. Si

nous calculons un descripteur de type HOG sur les deux images, les gradients auront la même orientation et à peu près la même norme mais une direction opposée.

Afin de gérer les inversions de gradient, nous avons restreint un descripteur HOG traditionnel à une demi-taille dont les directions de gradient opposées sont additionnées (Equations (2.5) et (2.6)). Ce concept est illustré dans la figure 2.4 déjà présenté dans la partie 2.1.2 : de cette façon, nous conservons dans le descripteur les informations d'orientation de gradient sans information de direction.

$$h_i = \sum_k \alpha_{\theta_k} r_k \tag{2.5}$$

$$\alpha_{\theta_k} = \begin{cases} 1 & \text{si } \theta_k \in [\frac{i}{N}\pi, \frac{i+1}{N}\pi] \cup [\frac{i}{N}\pi + \pi, \frac{i+1}{N}\pi + \pi] \\ 0 & \text{sinon} \end{cases} \tag{2.6}$$

avec N, le nombre choisi d'intervalles dans l'histogramme, h_i le i^{me} intervalle, θ_k l'orientation du gradient au pixel k et r_k l'amplitude du gradient au pixel k. Le coefficient α_{θ_k} est égal à 1 lorsque l'orientation du gradient au pixel k est inclue dans l'intervalle $[\frac{i}{N}\pi, \frac{i+1}{N}\pi]$ ou dans sa direction opposée (en $[\frac{i}{N}\pi + \pi, \frac{i+1}{N}\pi + \pi]$).

Application du noyau de Hellinger sur les descripteurs

Il a été prouvé dans plusieurs études que l'utilisation de la distance euclidienne n'est pas la meilleure pratique pour comparer les caractéristiques qui portent des informations sous forme d'histogrammes. Pour ces cas particuliers, χ^2 ou la métrique de Hellinger sont de meilleurs choix. Cette considération faite, les auteurs de [AZ12] proposent quelques modifications sur le descripteur SIFT et l'appellent RootSIFT. L'idée derrière cette évolution est que la comparaison des descripteurs RootSIFT avec l'aide de la distance Euclidienne est la même que l'application du noyau Hellinger sur les descripteurs SIFT originaux. Nous appliquons le même processus sur nos histogrammes de gradients : soit $\mathbf{h}_1$ et $\mathbf{h}_2$ deux vecteurs unitaires selon la norme euclidienne ($\|\mathbf{h}_i\|_2 = 1$), Leur distance euclidienne est donnée par les equations (2.7) et (2.8) :

$$d_{Eucl}(\mathbf{h}_1, \mathbf{h}_2) = \|\mathbf{h}_1 - \mathbf{h}_2\|_2 = \sqrt{\|\mathbf{h}_1\|_2^2 + \|\mathbf{h}_2\|_2^2 - 2{\mathbf{h}_1}^T\mathbf{h}_2} \tag{2.7}$$

$$d_{Eucl}(\mathbf{h}_1, \mathbf{h}_2) = \sqrt{2 - 2K_{Eucl}(\mathbf{h}_1, \mathbf{h}_2)} \tag{2.8}$$

où $K_{Eucl}(\mathbf{h}_1, \mathbf{h}_2) = {\mathbf{h}_1}^T\mathbf{h}_2$ est le noyau Euclidien (ou similarité). Nous souhaitons remplacer cette similartié par le noyau de Hellinger donné dans l'équation 2.9 :

$$K_{Hell}(\mathbf{h}_1, \mathbf{h}_2) = \sum_{j=1}^{N} \sqrt{h_{1j} h_{2j}} \tag{2.9}$$

pour $\mathbf{h}_1$ et $\mathbf{h}_2$ deux histogrammes normalisés selon $L1$ ($\sum_{j=1}^{N} h_{ij} = 1$ et $h_{ij} \geq 0$). Un moyen simple de calculer la similitude de Hellinger sur les descripteurs est de normaliser les vecteurs d'histogrammes et de passer à la racine carrée chaque élément des histogrammes. Ainsi, $K_{Eucl}(\sqrt{\mathbf{h}_1}, \sqrt{\mathbf{h}_2}) = \sqrt{\mathbf{h}_1}^T \sqrt{\mathbf{h}_2} = K_{Hell}(\mathbf{h}_1, \mathbf{h}_2)$ et l'utilisation de la distance Euclidienne sur ces descripteurs modifiés équivaut à utiliser la similitude de Hellinger sur les descripteurs initiaux.

Requêtes par «Sac-de-mots»

Une stratégie courante utilisée pour la recherche d'images par le contenu est de calculer un «sac de mots» (*Bag-of-Words, BoW*) comme expliqué dans [SZ03]. Les données utilisées sont divisées en deux ensembles : la première compose la mémoire et la seconde est appelée séquence *live* ou «en ligne» car cette information est généralement acquise progressivement pendant le processus de localisation. La recherche *via* une approche *BoW* tire profit d'une étape de prétraitement appliquée sur les images de la partie mémoire de l'ensemble des données. Un schéma global du processus *Bag-of-Words* est présenté dans la figure 2.26. Toules les caractéristiques locales sont d'abord extraites de toutes les images de la mémoire. Un algorithme *K-means* sépare alors l'espace entier des descripteurs selon K clusters (de 1000 à 8000 clusters selon les différents cas de test). L'ensemble des descripteurs moyens de chaque cluster est décrit comme le *vocabulaire*. Tous les descripteurs sont ensuite quantifiés par rapport à ce vocabulaire. Chaque *mot* du vocabulaire est ensuite pondéré par un score *TF-IDF* (*Term Frequency-Inverse Document Frequency*). Le score TF-IDF est le produit de deux termes :

— $\text{tf}_{i,j}$ (*Term Frequency* ou la «fréquence») est défini comme tel (équation (2.10)) :

$$\text{tf}_{i,j} \triangleq \frac{n_{i,j}}{\sum_{k=1}^{|I|} n_{k,j}} \tag{2.10}$$

où $n_{i,j}$ est le nombre d'occurences du mot d'index i dans le dictionnaire, dans l'image d'index j dans la séquence d'images composant la mémoire, et $|I|$ est le nombre total d'images dans le *corpus*.

— idf_i (*Inverse Document Frequency* ou «fréquence inverse de document») est défini ainsi (équation (2.11)) :

$$\text{idf}_i \triangleq \log \frac{|I|}{|\{i_j : w_i \in i_j\}|} \tag{2.11}$$

où $|I|$ est le nombre total d'images dans le corpus et $|\{i_j : w_i \in i_j\}|$ est le nombre d'images dans lesquelles apparaissent le mot d'index w_i.

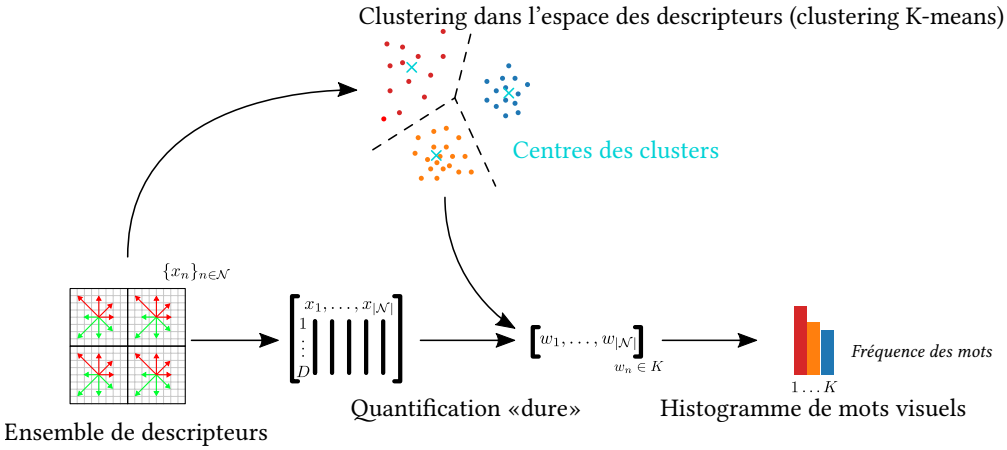


FIGURE 2.26 – Représentation graphique d'une approche *Bag-of-Words* : un *ensemble d'apprentissage* est d'abord utilisé afin d'agréger les caractéristiques extraites dans l'espace de description. Les caractéristiques sont ensuite quantifiées selon le *cluster* le plus proche. La représentation résultante des images est un «histogramme de mots visuels».

Ainsi, les scores $\text{tf-idf}_{i,j} = \text{tf}_{i,j} \times \text{idf}_i$ permettent de spécifier les mots qui sont les plus pertinents à la fois dans une image donnée de la mémoire et dans toute la base de données. Pendant le traitement de la séquence «en-ligne», nous extrayons des caractéristiques de chaque image et les quantifions par rapport au dictionnaire calculé avec la séquence en mémoire. Les mots les plus pertinents trouvés dans l'image courante décrivent l'image la plus proche présente en mémoire.

2.3.2 PHROG appliqué à la problématique de la localisation visuelle

Comme nous l'avons mentionné précédemment, plusieurs travaux et développements ont été réalisés pour faire face au problème de la reconnaissance visuelle de lieux à long terme. Ces travaux se concentrent sur deux approches principales : la première concerne la description de l'image elle-même et propose des études pour améliorer l'appariement d'images une à une. L'autre approche considère une séquence et pas seulement une image. Ainsi, la cohérence temporelle permet d'envisager un filtrage afin d'éliminer de potentiels appariements inappropriés. Les travaux présentés dans cette partie se concentrent sur la première approche, nous aborderons la seconde approche dans la partie suivante. Notre objectif ici est d'améliorer l'appariement de

deux images de la même scène lorsque des capteurs avec différentes plages spectrales sont utilisés. Afin d'évaluer notre méthode et les méthodes bien connues dans la littérature, nous utilisons plusieurs ensembles de données. Certains proviennent de travaux récents disponibles dans la communauté [BS11, MPM15, AAS+16]. Nous avons préparé notre propre ensemble de données supplémentaires pour évaluer les différentes techniques face à des contraintes et difficultés accrues.

Pour chaque cas de test, nous prenons une seule modalité à partir d'un jeu de données (visible ou infrarouge) et construisons un *codebook* hors ligne. Nous désignons ce sous-ensemble comme «mémoire». Après quelques expérimentations, nous avons déterminé que les dictionnaires composés de 1000 mots sont un bon compromis entre efficacité et vitesse de calcul. Nous désignons l'autre sous-ensemble comme séquence «en ligne» (ou *live*). Il est composé d'images de l'autre modalité composant l'ensemble de données. En d'autres termes, pour chaque cas de test, le dictionnaire est construit grâce à une seule modalité et l'étape de requête est toujours effectuée avec une autre modalité. Nous essayons de faire correspondre chaque image composant le sous-ensemble dit «live» en trouvant l'image la plus proche dans la mémoire en fonction des histogrammes de mots dans chaque image. Si l'image de la mémoire retournée par l'algorithme et l'image en direct proviennent de la même paire, nous considérons le test comme un «vrai positif», sinon comme un «faux positif». Nous expérimentons cette méthode sur les cas de test présentés dans les parties suivantes. Nous donnons pour chacun le ratio entre les vrais positifs et le nombre total d'images en direct dans l'ensemble de données. Nous montrons également avec la figure 2.27 un exemple où notre algorithme proposé échoue en raison de l'*aliasing* dominant entre les deux scènes (forme du chemin de fer, bâtiment sur le côté droit, arrière-plan avec des montagnes, *etc*).

Aliasing

Configuration matérielle des expérimentations

Nous avons mené les expérimentations sur un ordinateur de bureau exécutant *Ubuntu 16.04 LTS* avec un processeur *Intel Core i7* et 8 GiOctets de RAM. Nous avons limité à 10000 le nombre de points caractéristiques détectés dans chaque image. Nous avons donné des détails sur le motif de description dans les sections précédentes (Figure 2.25) et la modification d'un descripteur type *HOG*. Un vecteur de description PHROG est donc 2 fois plus petit (64 scalaires) que celui de SIFT. Avec de tels paramètres, le calcul d'un dictionnaire de 1000 mots visuels dure entre une heure et deux heures selon l'ensemble de données considéré.

FIGURE 2.27 – Un exemple de faux appariemment lorsque l'algorithme proposé échoue. On remarque que la confusion (*aliasing*) entre la requête (image de gauche) et l'image de la mémoire retournée par l'algorithme est forte.

Expérimentations sur des images visibles et infrarouge proche

Le premier ensemble de données provient de l'EPFL (*École Polytechnique Fédérale de Lausanne* en Suisse) et est présenté dans [BS11]. Il est composé de plusieurs sous-groupes triés : *country*, *field*, *forest*, *indoor*, *mountain*, *oldbuilding*, *street*, *urban* et *water*. Chaque sous-ensemble est composé d'environ 50 paires d'images. Chaque paire comprend une image visible et une contrepartie dans l'infrarouge proche (NIR). Les images de chaque paire ont été corrigées par les auteurs afin que les points de vue et les résolutions soient strictement identiques. Un exemple de paire est donné dans la Figure 2.28. Nous choisissons de nous concentrer sur les sous-ensembles *urban*, *street* et *country* qui sont plus proches des cas d'utilisation de la robotique et de la navigation. Le sous-ensemble *country* a effectivement été utilisé dans [AAS+16] comme l'ensemble d'apprentissage pour l'ensemble des expérimentations. Nous choisissons alternativement les images NIR et l'ensemble visible comme mémoire et son opposé en direct.

Les résultats sont donnés dans le tableau 2.1 et figures 2.29 et 2.31 pour chaque sous-ensemble et associations détecteur-descripteur et nous montrons un exemple de matrice de confusion obtenue sur l'ensemble de données *urban* avec PHROG (figure 2.32). Nous pouvons voir facilement que la diagonale de la matrice de confusion présente les distances les plus faibles entre les images : les jeux de données ont été synchronisés afin de calculer les performances de manière simple. Cela nous permet de considérer que les images associées à la diagonale sont les bons appariements et que les autres sont des distances calculées pour des correspondances fausses. Cette configuration des matrices de confusion rend possible le calcul des courbes *Precision-*

Recall (*PR*) et leur AUC.

FIGURE 2.28 – Une paire visible-infrarouge proche issue du jeu de l'EPFL.

	Jeux de données					
	Urban		**Street**		**Country**	
SIFT-SIFT	96%	94%	78%	66%	40%	34%
SIFT-GISIFT	98%	94%	70%	64%	34%	32%
FAST-SIFT	100%	100%	96%	96%	75%	76%
Harris-SIFT	100%	98%	88%	92%	73%	61%
Harris-GISIFT	100%	100%	90%	90%	69%	67%
Harris-PHROG	100%	100%	96%	94%	73%	80%

TABLE 2.1 – Taux de bon appariements sur le jeu visible-infrarouge proche.

Comme on peut le voir, les images NIR ne sont pas si différentes des images visibles. Il semble que la recherche d'images par le contenu avec des fonctions traditionnelles donne de très bons rapports de correspondance (*cf* les ensembles de données *urban* et *street*). Néanmoins, les résultats sur l'ensemble de données *country* sont plus disparates : une explication possible est que la végétation est beaucoup plus présente dans cet ensemble de données et, par conséquent, le problème d'inversion du sens des gradients apparaît plus fréquemment qu'avec des bâtiments et matières inorganiques. Les courbes PR montrent que PHROG est le meilleur sur les ensembles de données *urban* et *street* : son AUC est plus élevé d'au moins 9% que les autres méthodes évaluées. Sur l'ensemble de données *country*, PHROG n'obtient pas le meilleur AUC, mais sa précision reste la meilleure lorsque le rappel est faible. Ce résultat est significatif car cela veut dire que si le processus de recherche renvoie un seul résultat, PHROG donne la meilleure réponse.

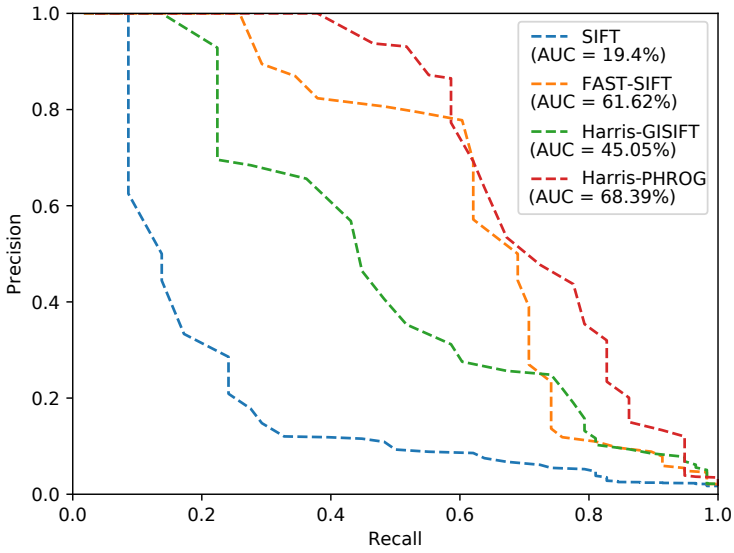


FIGURE 2.29 – Courbes Precision-Recall et leur AUC respective obtenues pour chaque méthode sur le jeu de données *urban*.

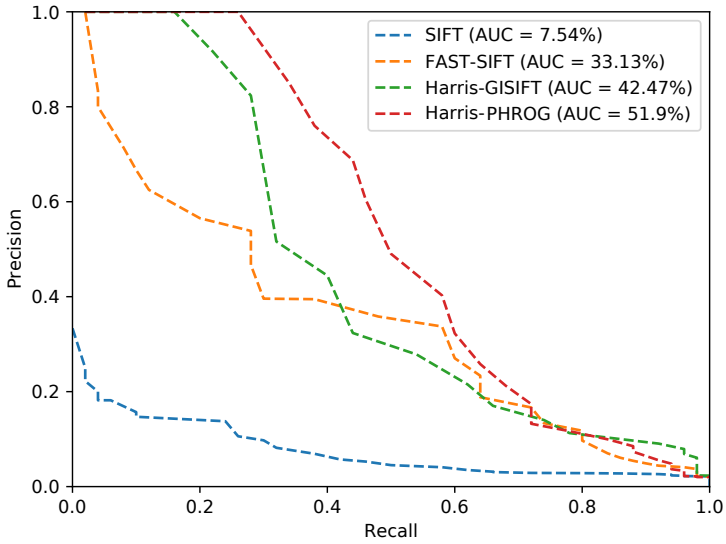


FIGURE 2.30 – Courbes Precision-Recall et leur AUC respective obtenues pour chaque méthode sur le jeu de données *street*.

Expérimentations sur des images visibles et infrarouge lointain

Cette partie considère des images infrarouges provenant d'une gamme spectrale beaucoup plus éloignée du spectre visible que la précédente. Nous avons utilisé l'en-

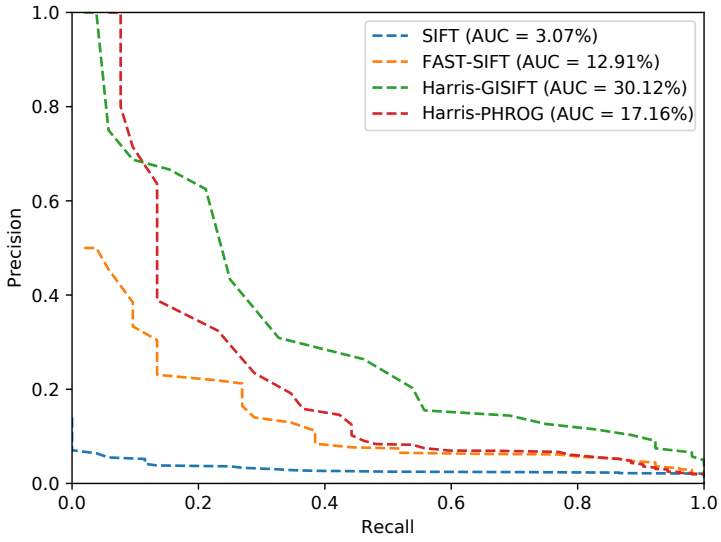


FIGURE 2.31 – Courbes Precision-Recall et leur AUC respective obtenues pour chaque méthode sur le jeu de données *country*.

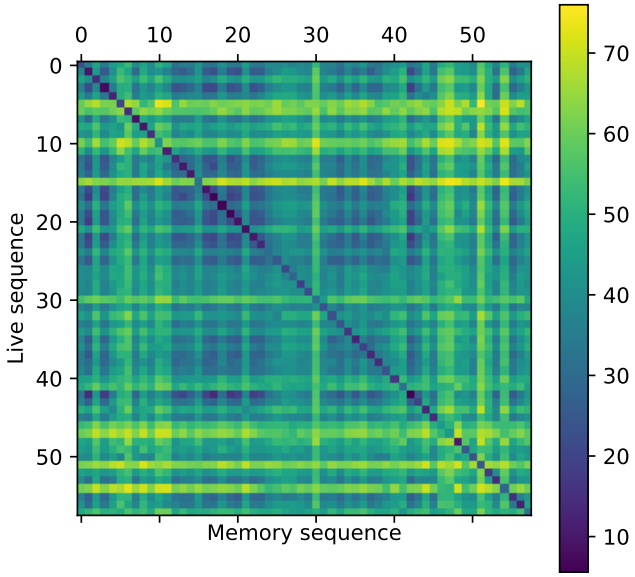


FIGURE 2.32 – Matrice de confusion entre la séquence mémoire et la séquence *live* obtenue avec PHROG. Les valeurs dans la matrice correspondent aux distances calculées pour chaque paire d'images possible.

semble de données infrarouge-visible introduit dans [AAS+16, AST15]. Cette gamme spectrale est appréciée pour sa réponse thermique et son utilisation possible en tant que système de vision nocturne. Cet ensemble de données englobe des scènes de vues extérieures du campus de Barcelone. Les images de cet ensemble de données ont également été corrigées par leurs auteurs afin que les résolutions et les points de vue soient identiques. Un exemple de l'ensemble de données se trouve en figure 2.33.

Nous expérimentons deux situations en changeant la modalité utilisée comme mémoire. Les résultats sont résumés dans le tableau 2.2 et la figure 2.34 présente les courbes PR. Nous pouvons constater que le ratio d'appariement est légèrement meilleur lorsque le sous-ensemble LWIR est utilisé comme source de la composition du dictionnaire. Nous supposons que les images moins texturées améliorent le calcul du dictionnaire des mots visuels. En outre, notre descripteur PHROG démontre sa valeur sur cet ensemble de données. PHROG donne des résultats nettement meilleurs que les autres méthodes avec exactement les mêmes paramètres que sur l'ensemble de données EPFL. Cependant, nous pouvons noter que les AUC sont très basses (inférieures à 5%), même si la précision de PHROG est bonne lorsque le rappel est faible. Cela signifie que les distances entre les images pour les correspondances vraies et fausses sont très proches l'une de l'autre mais assez discriminantes lorsqu'il est nécessaire de récupérer une seule image. Nous remarquons ainsi que plus les plages spectrales sont éloignées, plus il est difficile d'apparier les images avec des méthodes courantes de l'état de l'art en recherche d'image par le contenu.

FIGURE 2.33 – Une paire visible-infrarouge lointain issue du jeu de l'université de Barcelone.

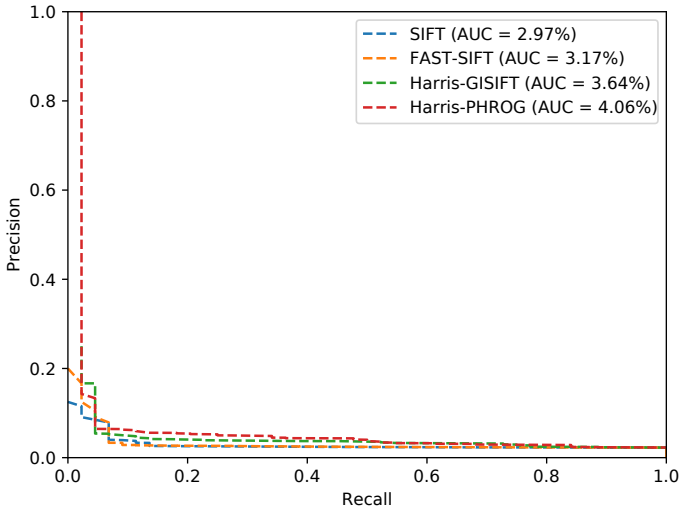


FIGURE 2.34 – Courbes Precision-Recall et leur AUC respective obtenues pour chaque méthode sur le jeu de données de l'université de Barcelone.

	Modalité utilisée en mémoire	
Méthodes	**LWIR**	**Visible**
SIFT-SIFT	9%	9%
SIFT-GISIFT	11%	9%
FAST-SIFT	22%	9%
Harris-SIFT	18%	20%
Harris-GISIFT	52%	38%
Harris-PHROG	61%	56%

TABLE 2.2 – Taux de bons appariements sur le jeu de l'université de Barcelone.

Expérimentations sur le jeu de données VPRiCE

Dans cette partie, nous utilisons un ensemble de données qui est composé d'images sélectionnées parmi l'ensemble de données VPRiCE [2] ; Un exemple de cet ensemble d'images est donné dans la figure 2.35. Il a été conçu pour évaluer la reconnaissance de lieux à long terme. Les deux séquences de l'ensemble de données ont été prises en utilisant des capteurs sensibles dans le spectre visible, mais à deux moments différents, de sorte que les changements saisonniers sont prépondérants. Par rapport aux deux jeux de données précédents, les images ne sont pas rectifiées de sorte que deux

2. https ://roboticvision.atlassian.net/wiki/pages/viewpage.action ?pageId=14188617

images de la même paire ont un point de vue différent. [NSP13, MPM15] présentent des résultats intéressants sur cet ensemble de données, mais ne donnent pas de détails sur les performances des extracteurs de caractéristiques sur des images simples : ils n'introduisent qu'une précision moyenne de leur méthode respective sur l'ensemble des séquences correspondant. Nous appliquons le même protocole d'évaluation sans approche séquentielle sur cet ensemble de données et obtenons les résultats présentés dans le tableau 2.3 et Figure 2.36.

FIGURE 2.35 – Une paire d'images issue du jeu VPRiCE.

Méthodes	**Efficacité**
SIFT-SIFT	36%
SIFT-GISIFT	42%
FAST-SIFT	68%
Harris-SIFT	52%
Harris-GISIFT	47%
Harris-PHROG	73%

TABLE 2.3 – Taux de bons appariements sur le jeu de données VPRiCE.

Même si cet ensemble de données est très difficile à appréhender, la caractéristique PHROG présente un appariement faux une fois sur quatre et surpasse les autres méthodes que nous avons testées. PHROG présente la meilleure AUC et la meilleure précision lorsque le rappel est faible.

Expérimentations sur le jeu de données visible-SWIR

Le dernier cas de test implique notre propre ensemble de données. Il a été réalisé avec une caméra visible et un capteur SWIR. Cet ensemble de données est beaucoup

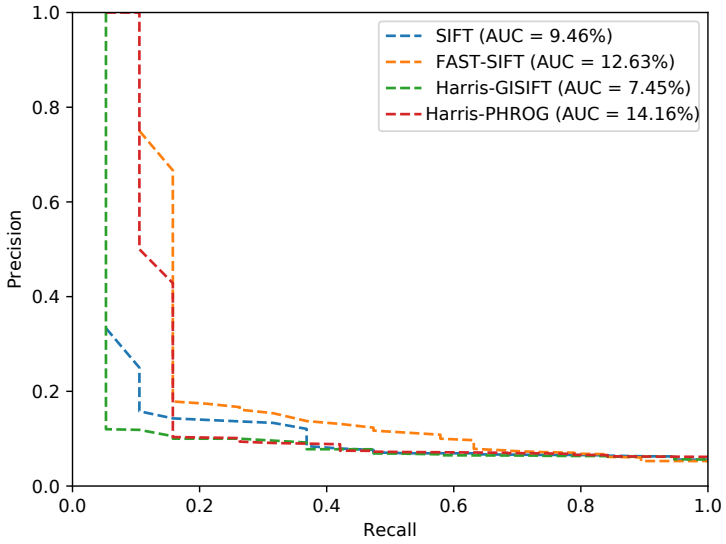


FIGURE 2.36 – Courbes Precision-Recall et leur AUC respective obtenues pour chaque méthode sur le jeu de données VPRiCE.

plus difficile car la résolution et les points de vue des images ne sont pas identiques. De plus, un sous-ensemble a été acquis plusieurs mois plus tard, de sorte que l'aspect de la végétation est très différent (avec et sans feuilles). La figure 2.37 est tirée de cet ensemble de données. L'ensemble SWIR, avec la plus petite résolution, a été utilisé comme mémoire en premier et les deux modalités ont été échangées par la suite. Les résultats sont présentés dans le tableau 2.4 et figure 2.38. Notre méthode montre encore de bons résultats par rapport aux descripteurs habituels. Une remarque notable doit être faite concernant le choix de la mémoire : les résultats sont meilleurs lorsque le SWIR compose la mémoire plutôt que l'inverse. Nous supposons que la résolution faible et sujette aux bruits de la caméra SWIR conduit à des descripteurs moins informatifs et à des représentations plus générales qu'avec l'ensemble visible. Néanmoins, ces résultats sont médiocres en termes absolus, les AUC sont très faibles et aucune méthode n'entraine de résultats significatifs si l'on considère uniquement les courbes Précision-Rappel. Nous avons effectué les tests avec cette caméra SWIR afin d'éprouver notre méthode dans une situation très contrainte. De toute évidence, cette caméra n'est pas le meilleur choix à faire lors de la conception d'un système embarqué, en raison de sa faible résolution par rapport à d'autres appareils, le bruit généré sur les images et son faible intérêt pour les situations de faible luminosité. En effet, le problème de reconnaissance visuelle de lieux nécessite un processus supplémentaire comme un filtrage pour affiner la correspondance avec plusieurs images entrantes par

exemple.

	Modalité utilisée en mémoire	
Méthodes	**SWIR**	**Visible**
SIFT-SIFT	15%	5%
SIFT-GISIFT	20%	10%
FAST-SIFT	15%	20%
Harris-SIFT	15%	10%
Harris-GISIFT	25%	15%
Harris-PHROG	35%	15%

TABLE 2.4 – Taux de bons appariements sur notre jeu de données (multimodal à long terme).

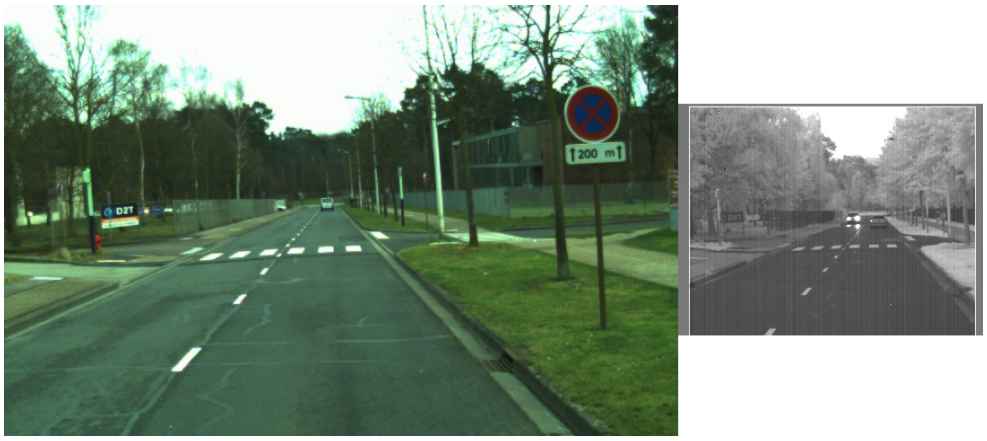


FIGURE 2.37 – Une paire visible-SWIR issue de notre jeu de données.

2.3.3 Discussion

Les capteurs visions ont été étudiés et évalués depuis des années en robotique, notamment pour les applications en extérieure et la navigation autonome. Les études et les implémentations ont été menées avec plusieurs types de caméras dont la sensibilité spectrale diffère. Chaque gamme spectrale apporte ses forces et ses faiblesses, ce qui se traduit par des choix techniques réalisés selon le but final : par exemple, une caméra couleur pour la reconnaissance des signes routiers, une approche proche infrarouge pour améliorer la perception pendant les journées brumeuses et l'infrarouge thermique pour la détection des piétons ou même la localisation visuelle pendant la

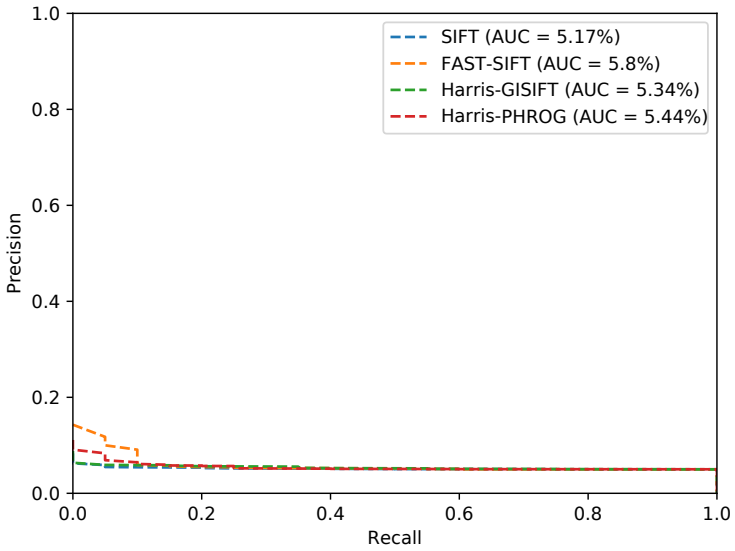


FIGURE 2.38 – Courbes Precision-Recall et leur AUC respective obtenues pour chaque méthode sur notre jeu de données.

nuit. En raison de ces considérations, il existe une diversité de systèmes de traitement adaptés à différents types de données issus de capteurs. Dans la poursuite de systèmes coopérants partageant leurs informations collectées dans l'environnement proche, il existe un besoin de concevoir des méthodes suffisamment générales et robustes aux changements de bande spectrale.

Alors que peu de travaux mettent l'accent sur ces problèmes, nous avons considéré dans cette section les contraintes introduites par le besoin d'invariance au changement de capteur, en particulier lorsque les systèmes robotiques intègrent des caméras avec différentes bandes spectrales. Nous avons étudié les approches de recherche d'image par le contenu entre un ensemble de données visible et infrarouge, dans les gammes spectrales NIR, SWIR ou LWIR. Nous avons fait ces expérimentations avec les contraintes typiques étudiées dans la littérature à long terme sur la reconnaissance des lieux visuels : les changements de points de vue et de résolution d'image, l'évolution de l'apparence à long terme et les variations d'illumination de la scène. Le travail développé dans cette section propose une nouvelle méthode de description des caractéristiques : sur la base d'une étude de plusieurs méthodes, nous avons déterminé que les détecteurs de caractéristiques traitant de formes contrastées comme Harris sont plus appropriés à une répétabilité de modalité croisée. Nous avons proposé un modèle de description avec une fonction HOG modifiée qui maximise la répétabilité dans différentes plages spectrales. Cette fonctionnalité est ensuite extraite à plusieurs

échelles afin de faire face aux changements de point de vue.

Nous avons évalué notre proposition sur plusieurs jeux de données bien connus utilisés pour l'analyse comparative des caractéristiques multimodales et un dédié à la reconnaissance visuelle. Nous avons proposé un autre ensemble de données avec un capteur SWIR dont les attributs (résolution, par exemple) sont plus difficiles à exploiter. De cette façon, notre étude présente des résultats expérimentaux sur quatre types de caméras avec différentes sensibilités spectrales, en veillant à ce que notre proposition ne corresponde pas à un couple particulier de capteurs visibles/IR et soit plutôt généraliste. Nous avons remarqué que certaines méthodes peuvent être suffisamment discriminantes avec une approche Bag-of-Word et sont pertinentes pour la localisation à long terme. Néanmoins, l'association multimodale nécessite encore des améliorations, en particulier lorsque les séquences à associer sont plus difficiles en raison de l'augmentation successives des contraintes (changements perceptuels, gammes spectrales lointaines, résolutions différentes, bruit, etc.). Notre proposition montre des résultats intéressants qui surpassent les extracteurs de caractéristiques traditionnels. Les caractéristiques locales demeurent une solution intéressante pour la mise en pratique d'une solution de *SLAM* : combiné avec des méthodes de stockage et de récupération de données (à savoir *Bag-of-Words*), on peut envisager d'utiliser les mêmes données d'une part pour la localisation globale et la détection de fermeture de boucle comme pour une tâche d'estimation de pose et de reconstruction 3D.

Conclusion

En fin de premier chapitre, nous avions évoqué la difficulté d'associer et comparer des informations visuelles de sources différentes, en particulier lorsque les bandes spectrales sont sensiblement éloignées. Dans ce chapitre, nous avons proposé une nouvelle méthode de description globale des images apportant une première réponse au problème abordé. Dans un second temps, nous nous sommes intéressés à la possibilité d'utiliser une méthode de description locale. Nous avons constaté que la détection de points d'intérêt était en elle-même problématique : en effet, lors du choix d'un détecteur et de ses paramètres, il faut veiller à maximiser la répétabilité entre deux modalité. Nous avons par la suite proposé un nouveau descripteur ponctuel que nous avons nommé PHROG. Celui-ci fait l'objet d'une publication dans le journal *Sensors*. Nos expérimentations tendent à montrer que celui-ci apporte de meilleurs résultats à la problématique de la localisation visuelle à long terme face aux autres méthodes de l'état de l'art, au détriment d'une utilisation en mémoire et en temps de calcul plus importante que ses concurrents.

Nous avons abordé jusqu'ici le problème en considérant les images une-à-une et sans *a priori*. La plupart du temps en robotique mobile, le système acquiert pourtant des images au fur et à mesure de sa progression. Les images successives ont donc logiquement un lien et décrivent, au moins en partie, la scène ou une scéne proche de celles representées dans les images précédentes. On peut parler ainsi de cohérence temporelle. Cette considération nous permet de considérer une séquence d'image plutôt qu'une image seule, et donc de confirmer ou infirmer des suppositions de localisation faites au cours de la séquence observée. Ces méthodes font l'objet du chapitre suivant.

CHAPITRE

3

LOCALISATION ET COHÉRENCE TEMPORELLE

Sommaire

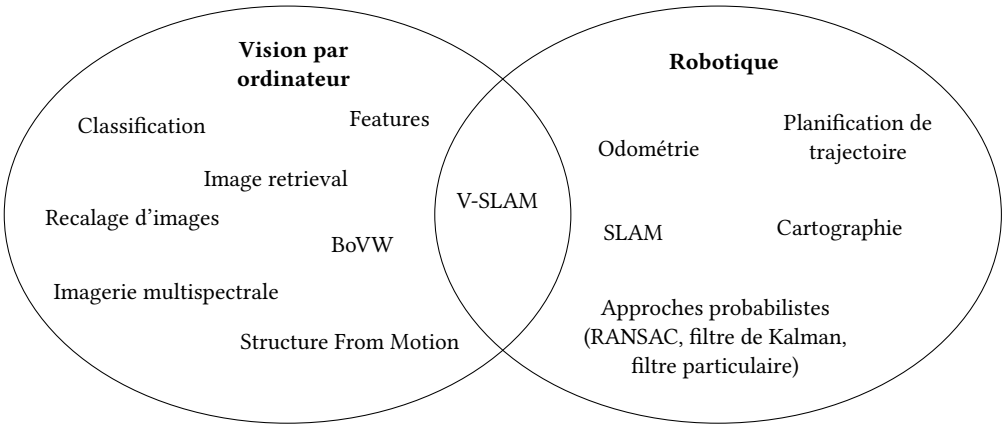


FIGURE 3.1 – Représentations des domaines de connaissance issus des disciplines de la vision par ordinateur et de la robotique et leurs interactions.

Les méthodes proposées et développées dans le chapitre 2 font appel à des connaissances et techniques issues des travaux menés ces dernières décennies dans le domaine du traitement image et de la vision par ordinateur. En parallèle, la communauté des chercheurs en robotique a proposé des méthodes implémentées au cœur de systèmes divers, de complexité variable et avec un grand nombre de capteurs différents. Les interactions entre les deux disciplines sont fortes et l'on peut représenter les méthode de chacune sous la forme présentée en figure 3.1.

SLAM

Les approches *Structure From Motion* d'une part et *SLAM* d'autre part sont un exemple de travaux très similaires approchés dans les deux disciplines. Pour les méthodes dites *Structure From Motion*, il s'agit de déduire un modèle 3D de l'environnement sous la forme d'un nuage de points ou de primitives denses éventuellement texturés à partir d'un ensemble d'images prises selon différents points de vue. Les méthodes de SLAM ont une approche similaire pour la tâche de cartographie (bien que les capteurs utilisés peuvent être de natures différentes). Néanmoins, alors que l'emphase est mise sur la qualité du modèle de l'environnement généré pour les méthodes relevant du domaine *Structure From Motion*, les approches *SLAM* mettent en avant les performances de localisation. Les méthodes de SLAM visuel (*V-SLAM*) faisant usage principalement de capteurs basés vision se trouvent donc à l'interface des deux disciplines.

Ces méthodes sus-citées considèrent non pas une mais plusieurs images pour recomposer une information spatiale. Lorsque ces images multiples proviennent d'une même séquence, on peut admettre comme hypothèse supplémentaire le fait qu'il y un lien entre les images successives : on parle de cohérence temporelle. L'objet

COHÉRENCE TEMPORELLE

de ce chapitre est de présenter brièvement dans une première section les méthodes utilisées en vision par ordinateur profitant de cette dimension temporelle. La deuxième section s'attelle à présenter les concepts et la mise en place d'un modèle probabiliste. Des résultats expérimentaux obtenus avec deux méthodes de filtrage différentes sont présentées dans une dernière partie.

3.1 Cohérence temporelle des séquences d'images

3.1.1 Asservissement visuel

Si l'on travaille avec des séquences images, et sous réserve que la fréquence d'acquisition est suffisamment élevée pour la tâche envisagée, on peut supposer que les phénomènes observés sont proches si bien que l'on peut mettre en place une méthode de suivi différentiel. C'est le cas par exemple de la méthode de Lucas-Kanade [LK+81] (que l'on qualifie parfois de *tracker* et que l'on croise donc sous l'acronyme *KLT*). Cette méthode suppose qu'un même point d'intérêt entre deux instants successifs a effectué un déplacement petit et que sa recherche peut s'effectuer dans un voisinage restreint. Ces approches permettent d'estimer le flux optique pour segmenter un objet en mouvement dans l'image ou estimer le changement de pose de la caméra.

Tracker

Flux optique

Un suivi différentiel peut également s'effectuer avec des descripteurs ou métriques s'appliquant à l'image entière. On regroupe généralement ces approches sous le nom d'«asservissement visuel». L'objectif dans ce cas va être de trouver la déformation appliquée (rigide ou non selon les choix des auteurs) afin de minimiser un coût calculé selon la similarité (comme l'information mutuelle) entre l'image initiale et l'image suivante déformée [DM10]. On retrouve dans ces approches des techniques et métriques courantes dans le domaine du recalage d'image en imagerie médicale, domaine qui a justement l'objectif d'associer des images issues de systèmes imageurs aux modalités très différentes. La robustesse de ces métriques face à la question de la multimodalité et leur coût computationnel faible en font des candidats idéaux pour un asservissement temps-réel entre images visibles et infrarouges [DM12] ou entre images synthétiques générées à partir des données d'un LIDAR et images visibles [WE14].

Asservissement visuel

3.1.2 Odométrie, *Structure From Motion* et *SLAM*

On rencontre d'autres méthodes itératives faisant usage de séquences d'images dans la littérature en robotique : la première concerne l'odométrie visuelle. Grâce à des méthodes d'association 2D-2D et les principes de la géométrie épipolaire princi-

palement [HZ03], on peut déterminer une pose (une orientation et un déplacement) entre deux positions d'une caméra à l'aide des images prises à ces deux instants. En cumulant les estimations de pose instants après instants, on peut en déduire une trajectoire effectuée par la caméra. Cette méthode d'odométrie visuelle couplée à une estimation de la structure par des approches 2D-3D permet de composer une représentation tridimensionnelle de l'environnement (*Structure From Motion*) voire continuer à évoluer dans cet environnement et poursuivre sa construction (*SLAM*).

3.1.3 *Robot kidnapping* et fermeture de boucle

Il existe des situations où une localisation du robot dans une carte vaste sans *a priori* est utile ou nécessaire. La littérature fait souvent référence à ce problème sous la dénomination *robot kidnapping*. Cela est utile si par exemple le système de localisation est défaillant un moment ou inutilisable temporairement à cause de conditions extérieures et continue à évoluer dans son environnement. Dans ce cas, une localisation sans condition initiale forte est nécessaire pour localiser à nouveau le robot et reprendre une méthode de suivi itérative.

Robot kidnapping

L'autre intérêt d'une localisation étendue dans une carte apparaît lorsque l'on implémente une solution de SLAM. En effet, comme nous l'avons vu, le robot dans ce cas évolue dans l'environnement, estime son déplacement, le plus souvent par odométrie, et compose la carte de l'environnement qu'il observe. L'odométrie étant sujette à une dérive plus ou moins prononcée, si le robot détecte qu'il navigue à nouveau dans une scène déjà observée (ce qui est permis par une localisation dans la carte composée depuis le début de sa progression), il peut mettre à jour la carte de manière à compenser la dérive et corriger les trajectoires enregistrées par une méthode d'«ajustement de faisceaux» (*bundle adjustment*). Cette tâche particulière en lien direct avec les méthodes de SLAM fait l'objet d'études dans de nombreux travaux de l'état de l'art. Les méthodes de «détection de fermeture de boucle» permettent de déterminer les scènes de l'environnement déjà parcourues et d'initier ainsi cet ajustement.

bundle adjustment

Fermeture de boucle

3.2 Mise en place d'un cadre probabiliste

Comme nous l'avions évoqué dans le premier chapitre de cette thèse, un robot est en interaction avec un environnement complexe et agit, ou adapte ses actions en fonction des contraintes auxquelles il est soumis. Lors de sa conception, il faut donc anticiper les potentielles situations rencontrées pour tenter de prévoir au mieux et pondérer l'imprévisible. De ce fait, les systèmes faisant appel à un cadre théorique

probabiliste sont courants et nous proposons dans cette partie deux approches y ayant également recours.

3.2.1 Probabilités et théorie Bayésienne

Notations de base et règle de Bayes

Dans les approches probabilistes, on représente les états (ou variables) d'un système par des variables aléatoires notées par exemple X. Une variable aléatoire X peut prendre une valeur spécifique notée x.

$$p(x) = p(X = x) \tag{3.1}$$

Une variable aléatoire peut être définie sur un espace discret ou continu (on parle alors de «densité de probabilité»). Dans les deux cas, la somme ou l'intégrale des probabilités de l'ensemble des évènements possibles est égale à 1 (équations (3.2) et (3.3)).

densité de probabilité

$$\sum_x p(x) = 1 \tag{3.2}$$

$$\int p(x)dx = 1 \tag{3.3}$$

On définit la «Distribution jointe» de deux variables aléatoires X et Y comme la distribution des réalisations conjointes sur l'ensemble des évènements possibles des deux variables (équation (3.4)).

Distribution jointe

$$p(x, y) = p(X = x \; et \; Y = y) \tag{3.4}$$

Deux variables aléatoires sont dites «indépendantes» si leur distribution jointe est égale au produit de leur distribution respective (équation (3.5)).

Variables aléatoires indépendantes

$$p(x, y) = p(x)p(y) \tag{3.5}$$

On nomme «probabilité conditionnelle» la probabilité qu'un évènement x se réalise sachant y (équation (3.6)).

probabilité conditionnelle

$$p(x|y) = \frac{p(x, y)}{p(y)} \tag{3.6}$$

Ainsi, si X et Y sont des variables aléatoires indépendantes, la probabilité conditionnelle de l'une sachant l'autre est égale à sa propre distribution de probabilité (équation (3.7)).

$$p(x|y) = \frac{p(x)p(y)}{p(y)} = p(x) \tag{3.7}$$

FORMULE DES PROBABILITÉS TOTALES

On peut déduire de ces dernières propriétés la «Formule des probabilités totales» ou *Theorem of total probability* dans le cas discret (équation (3.8)) de même que dans le cas continu (équation (3.9)).

$$p(x) = \sum_y p(x|y)p(y) \tag{3.8}$$

$$p(x) = \int p(x|y)p(y)dy \tag{3.9}$$

RÈGLE DE BAYES

De ces équations découle la «règle de Bayes» dans le cas discret (équation (3.10)) ainsi que dans le cas continu (équation (3.11)).

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{x'} p(y|x')p(x')} \tag{3.10}$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x')p(x')dx'} \tag{3.11}$$

La règle de Bayes est une modélisation intéressante d'un processus d'acquisition de données bruitées ou sujettes à erreurs. En effet, en supposant que x modélise un état (une propriété de l'environnement ou d'un robot par exemple) que l'on veut inférer d'une mesure y fournie par un capteur :

- $p(x)$ est l'information que l'on a de l'état x *a priori* (on parle de *prior probability distribution*) ;
- $p(y)$ représente les données issues de la mesure effectuée (*data*) ;
- la distribution de probabilité $p(y|x)$ modélise la relation entre l'état réel et l'information renvoyée par le capteur (*generative model*).

La probabilité $p(x|y)$ est ainsi appelée *posterior probability distribution.*

Prise en compte du temps

On inclut désormais la notion de temps dans les notations :

- La variable aléatoire x_t représente l'état du système considéré à l'instant t (ex : position d'un robot, vitesse, *etc*) ;
- La variable aléatoire u_t représente l'«action de contrôle» exercée par le système à l'instant t (ex : avancer, manipuler un objet de l'environnement, *etc*) ;
- La variable aléatoire z_t représente les «mesures» de l'environnement effectuées par les capteurs du système à l'instant t (ex : distance mesurée, rotation, *etc*).

On introduit alors une nouvelle notation permettant de décrire la probabilité jointe d'une variable alétoire réalisée sur un ensemble d'instants successifs (équation (3.12)).

$$p(x_{t_1:t_2}) = p(x_{t_1}, x_{t_1+1}, \ldots, x_{t_2}) \tag{3.12}$$

avec $t_1 \leq t_2$.

Dans le cas général avec les hypothèses établies jusqu'ici, la réalisation d'un état x_t dépend de tous les états, actions de contrôle et mesures passées (équation (3.13)).

$$p(x_t) = p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}) \tag{3.13}$$

3.2.2 Simplification du processus en chaîne de Markov

On pose comme hypothèse supplémentaire le fait que le processus de mesure n'a pas d'influence sur l'état du système. Ainsi l'état est «conditionnellement indépendant» de la mesure ce qui se traduit par l'équation 3.14.

$$p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t|x_{0:t-1}, u_{1:t}) \tag{3.14}$$

On suppose en outre que l'état est *complet*, autrement dit, l'état du système à l'instant t est uniquement stochastiquement dépendant de l'état du système à l'instant antérieur $t-1$ et de l'action de contrôle à l'instant t. Le système est ainsi modélisé par une «chaîne de Markov» (équation (3.15)).

Chaîne de Markov

$$p(x_t|x_{0:t-1}, u_{1:t}) = p(x_t|x_{t-1}, u_t) \tag{3.15}$$

De même, la mesure à l'instant t est uniquement stochastiquement dépendante de l'état du système à l'instant t (équation (3.16)).

$$p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t|x_t) \tag{3.16}$$

Ce modèle est connu sous les noms de «Modèle de Markov caché» (*hidden Markov model* ou HMM) ou «réseau bayésien dynamique» (*dynamic bayesian network*) que l'on peut réprésenter schématiquement selon la figure 3.2.

Définitions supplémentaires et algorithme de Bayes

La littérature définit de nouvelles distributions permettant de simplifier les notations utilisées dans l'élaboration des algorithmes bayésiens. On définit ainsi la *prediction* (équation (3.17)) comme étant l'estimation de l'état à l'instant t sachant les actions de commandes effectuées sans prise en compte de la dernière mesure acquise par le ou les

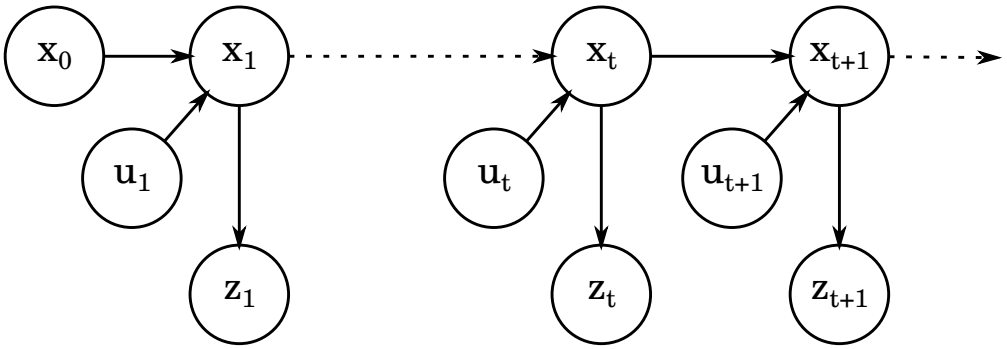


FIGURE 3.2 – Représentation graphique des dépendances stochastiques d'une chaîne de Markov.

capteurs. De même, on définit l'«état de connaissance», *belief*, *state of knowledge* ou encore *information state* selon les publications (équation (3.18)). Il s'agit de représenter l'état estimé du système à l'instant t sachant les actions de commandes effectuées et les dernières mesures acquises.

$$\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t}) \tag{3.17}$$

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \tag{3.18}$$

L'algorithme bayésien permet de calculer itérativement les estimations de l'état du système en prenant en compte à la fois le modèle d'évolution du système (étape de prédiction) et l'affinement de cette estimation par intégration des mesures effectuées par les capteurs. L'étape permettant de calculer $bel(x_t)$ à partir de $\overline{bel}(x_t)$ s'appelle *correction* ou *measurement update*. On peut alors donner la structure générale d'un algorithme bayésien avec les notations évoquées jusqu'ici (algorithme 1).

Algorithme 1 Définition générale d'un algorithme itératif Bayésien.

Données en entrée : $bel(x_{t-1})$, u_t, z_t
À l'instant t, pour tout état $x_t \in \Omega_X$:
1 : $\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1})bel(x_{t-1})dx_{t-1}$ ▷ Prediction
2 : $bel(x_t) = \eta p(z_t|x_t)\overline{bel}(x_t)$ ▷ Measurement update
avec η tel que $\int bel(x_t)dx_t = 1$
Données en sortie : $bel(x_t)$

3.2.3 Hypothèses supplémentaires et types de filtres

Lors de l'implémentation d'un filtre bayésien, il faut définir les distributions et espaces d'états possibles du système modélisé. En outre, il y a 3 hypothèses importantes à établir :

- *State transition probability* : $p(x_t|x_{t-1}, u_t)$. Cette distribution permet de modéliser l'évolution de l'état du système sachant son état antérieur et l'action de commande efffectuée ;
- *Measurement probability* : $p(z_t|x_t)$. Cette distribution modélise l'information issue du capteur sachant l'état réel du système ;
- *Initial belief* : $bel(x_0)$ (soit $p(x_0)$). Cette distribution correspond à la connaissance que l'on a du système à l'état initial.

L'algorithme bayésien défini par 1 est une définition générale. Il est possible de l'implémenter de différentes manières selon les hypothèses que l'on effectue sur la modélisation du système considéré et de ses interactions. En outre ces choix sont contraints par la possibilité d'appréhender la résolution mathématique des distributions mises en œuvre (en particulier l'étape de prédiction qui peut être difficile à calculer).

Plusieurs types d'implémentations ont ainsi été proposées dans l'état de l'art [TBF+05]. Les familles de filtres sont représentées dans la figure 3.3. Comme les connaissances en mathématiques nous permettent de manipuler aisément les distributions gaussiennes, quelques filtres manipulant ces densités de probabilités ont été proposés comme le filtre de Kalman ou encore le filtre de Kalman étendu. D'autres filtres, dits non-paramétriques, proposent différents type d'implémentation où les espaces des états possibles sont discrétisés, soit parce que le système est modélisé lui-même selon un espace discret (filtre bayésien discret), soit en discrétisant l'espace de définition continu (filtre d'histogrammes et filtres particulaires).

Les sections suivantes présentent deux implémentations liées à notre problème de localisation. La première met en œuvre un filtre de Bayes discret avec un modèle simple, fait appel au descripteur global présenté dans le deuxième chapitre et permet une localisation dans un espace discret, sans *a priori* sur la localisation initiale du système. La deuxième propose l'implémentation d'un filtre particulaire mettant en œuvre odométrie visuelle et l'approche PHROG proposée dans le chapitre 2. Cette méthode nécessite une localisation initiale approximative du système et s'apparente à un suivi de trajectoire du système. L'usage de PHROG conjointement au filtre particulaire permet de compenser partiellement les dérives engendrées par la méthode d'odométrie visuelle.

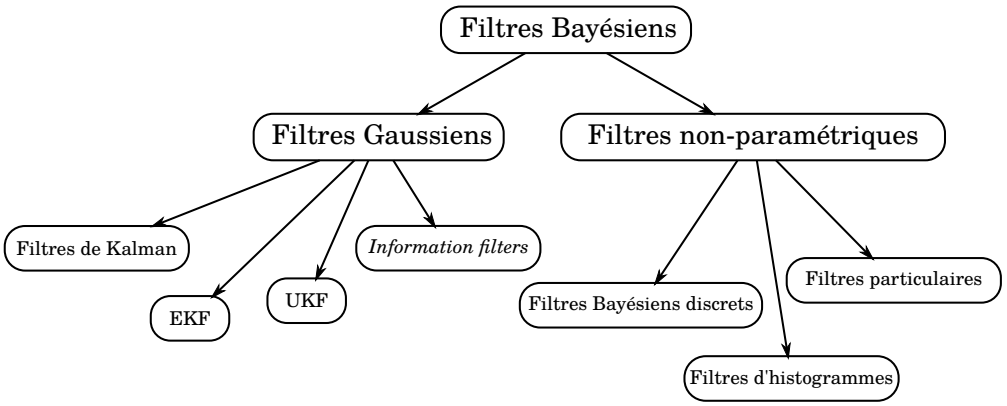


FIGURE 3.3 – Représentation des différentes familles d'implémentation des filtres bayésiens.

3.3 Implémentation de deux filtres probabilistes

3.3.1 Approche avec un filtre Bayésien discret

Un schéma global de la méthode mise en œuvre dans cette partie est visible en figure 3.4. Cette méthode considère chaque image de la séquence en mémoire comme une scène possible que notre système observe. Le nombre d'images disponibles en mémoire donne donc le nombre fini d'état possible du système.

Lorsque l'on a un nombre fini d'états, l'algorithme du filtre bayésien discret découle directement de la définition générale (algorithme 3).

Algorithme 2 Définition générale d'un algorithme bayésien discret.

Données en entrée : $\{p_{k,t-1}\}$, u_t, z_t
À l'instant t, pour tout état $x_t \in \Omega_X$ de probabilité $p_{k,t}$:

1 : $\overline{p}_{k,t} = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i) p_{i,t-1}$ ▷ Prediction
2 : $p_{k,t} = \eta p(z_t | X_t = x_k) \overline{p}_{k,t}$ ▷ Measurement update

avec η tel que $\sum_k p_{k,t} = 1$
Données en sortie : $p_{k,t}$

Mise en pratique

On choisit comme espace d'état Ω_X l'ensemble des index des images contenues dans la base de données préétablie. On définit alors $bel(x_t) = p(x_t | z_{1:t}, u_{1:t}) \in \Omega_X$ la probabilité que la scène observée à l'instant t soit la même que celle de l'image d'index

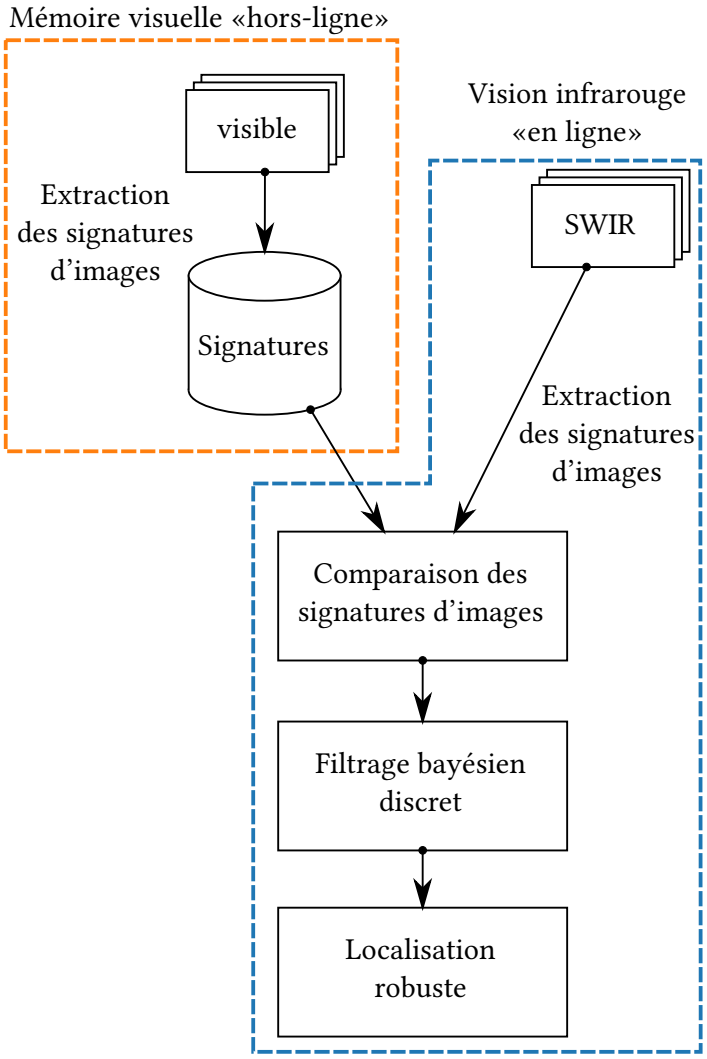


FIGURE 3.4 – Vue globale du système implémentant un filtre bayésien discret.

ω de la base de données.

$bel(x_0) = p(x_0)$ est définie comme une densité de probabilité équitablement répartie sur l'ensemble des index de la base de données (nous n'avons pas d'*a priori* sur la position initiale du véhicule). On définit les probabilités de transition d'état ainsi, pour tout $t > 0$ et tout $\omega \in \Omega_X$:

$$p(X_t = \omega + 1 | X_{t-1} = \omega) = \frac{1}{3}$$

$$p(X_t = \omega - 1 | X_{t-1} = \omega) = \frac{1}{3}$$

$$p(X_t = \omega | X_{t-1} = \omega) = \frac{1}{3}$$

Autrement dit, le véhicule à l'instant t parcourt probablement la même scène qu'à l'instant $t-1$, ou une scène concomitante de la base de données. On définit les probabilités de mesure ainsi, pour tout $t > 0$ et tout $\omega \in \Omega_X$:

$$p(Z_t = \omega | X_t = \omega) = 0,7$$

$$p(Z_t \neq \omega | X_t = \omega) = 0,3$$

Autrement dit, l'appariement image a une probabilité de 30% d'être faux.

On peut alors calculer la formule de prédiction (équation (3.19)) et de mise à jour (équation (3.20)).

$$\overline{p}_{k,t} = \frac{1}{3}(p_{k-1,t-1} + p_{k,t-1} + p_{k+1,t-1}) \tag{3.19}$$

$$\begin{cases} \tilde{p}_{k,t} = 0.7\,\overline{p}_{k,t} & \text{si meilleur appariement donné à l'indice k,} \\ \qquad\;\; = 0.3\,\overline{p}_{k,t} & \text{sinon.} \end{cases} \tag{3.20}$$

On peut ainsi en déduire facilement $p_{k,t}$ (équations (3.21) et (3.22)).

$$\eta = \frac{1}{\sum_k \tilde{p}_{k,t}} \tag{3.21}$$

$$p_{k,t} = \eta \tilde{p}_{k,t} \tag{3.22}$$

Résultats expérimentaux

La figure 3.5 donne un exemple simple de la façon dont nous traitons les résultats avec notre algorithme : les données en abscisse représentent les probabilités que l'image courante décrive le même endroit que l'image de la mémoire à l'index donné. Pour cet exemple, nous constituons une base de données avec seulement 20 images. Différents instants t apparaissent sur l'axe des ordonnées. Comme on peut le déduire du fait que les probabilités soient réparties uniformément selon toutes les images sur la première ligne, celle-ci correspond à l'étape initiale du filtre Bayésien.

Pendant la période de $t = 15$ à $t = 24$, nous n'avons pas d'image en mémoire avec une probabilité dominante : la tâche d'association échoue temporairement et converge à nouveau par la suite. Comme on peut le voir, une sorte de chemin prend forme dans la base de données.

Nous considérons trois possibilités différentes pour un jeu de test :

— Un chemin prend forme après quelques étapes et reste bien défini par la suite, la scène décrite par l'image provenant de la base de données dont la probabilité est la plus élevée correspond bien à la scéne observée ;
— Un chemin prend forme, mais l'algorithme diverge pendant un moment comme dans l'exemple donné par la figure 3.5, la localisation est temporairement perdue ;
— Aucun chemin ne prend forme, l'algorithme échoue ;

Nous avons constitué un ensemble de données avec environ 2000 images pour chaque modalité (visible et SWIR) prises à 10 Hz sur le toit d'une voiture, principalement dans la banlieue de Rouen, en France. Ces images ont été prises pendant des jours très différents. Nous avons sélectionné manuellement plusieurs sous-séquences avec des taux inférieurs et des chevauchements partiels. Nous avons créé une base de données représentant 100 endroits différents. Nous avons ensuite testé 20 séquences d'images, chacune composée de 50 images. Pour 11 d'entre elles, l'algorithme a convergé définitivement, a perdu temporairement la localisation pour 5 séquences et a échoué totalement 4 fois.

Nous avons développé dans cette section une solution de vision pour la localisation d'un système : à partir d'expériences antérieures, nous créons une base de données composée de signatures globales d'images dont la méthode est décrite dans la section 2.1. Ces patchs sont dimensionnés selon les paramètres géométriques donnés par la matrice de calibration habituelle. Une fois «en ligne», équipé d'une autre caméra, nous conservons la même approche pour extraire les signatures d'images. Nous comparons ensuite la base de données et les signatures en ligne, équilibrant les scores grâce à un filtre probabiliste.

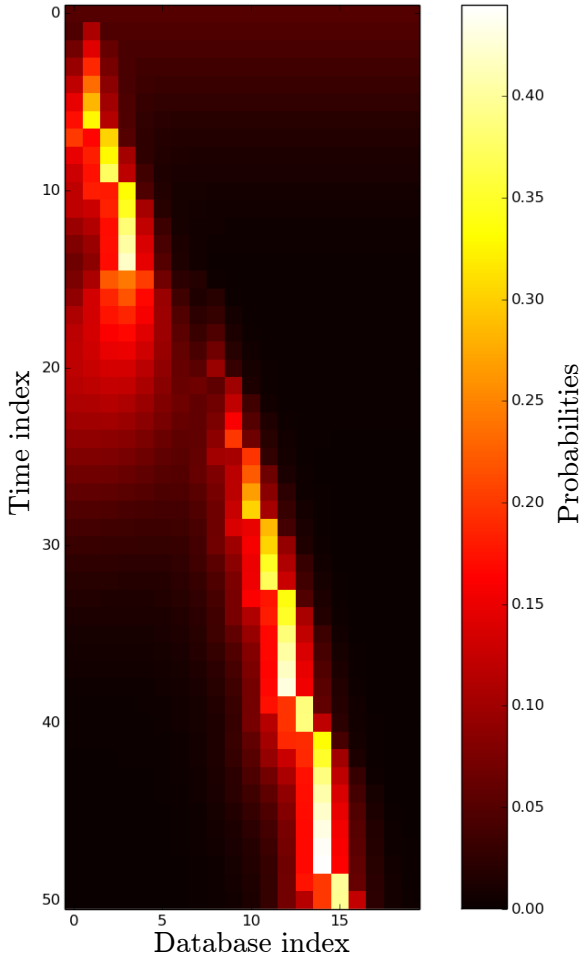


FIGURE 3.5 – Représentation graphique des scores de probabilité.

3.3.2 Filtre particulaire

Nous approchons dans cette partie un autre type de filtre non-paramétrique : le filtre particulaire. Contrairement au filtre bayésien discret que nous avons étudié dans la partie précédente ou encore au filtre d'histogramme pour lequel la discrétisation de la croyance est uniformément répartie sur l'intervalle de définition des états possibles du système, le filtre met à jour et conserve des hypothèses (les «particules») seulement pour les états de plus forte vraisemblance. Ainsi, la répartition des particules sera telle que les zones de l'espace d'état où la croyance est la plus forte sera peuplée de

nombreuses particules alors que les autres zones seront peu peuplées.

Un filtre représente donc la croyance $bel(x_t)$ par un ensemble Ψ_t de m particules $\psi_t^{[i]}$ à l'instant t (équation (3.23)).

$$\Psi_t \triangleq \left\{ \psi_t^{[1]}, \psi_t^{[2]}, \ldots, \psi_t^{[i]}, \ldots, \psi_t^{[m]} \right\} \tag{3.23}$$

Chaque particule est un tuple composé d'un vecteur d'état possible et d'un poids w_m (équation (3.24)).

$$\psi_t^{[i]} = \left\{ x_t^{[i]}, w_t^{[i]} \right\} = \left\{ \left[x_{1,t}^{[i]}, x_{2,t}^{[i]}, \ldots, x_{n,t}^{[i]} \right]^T, w_t^{[i]} \right\} \tag{3.24}$$

On définit alors l'algorithme du filtre particulaire comme suit :

Algorithme 3 Définition générale d'un algorithme de filtrage particulaire.

Données en entrée : $\{\Psi_{t-1}\}$, u_t, z_t
À l'instant t :
1 : $\overline{\Psi_t} = \Psi_t = \emptyset$
2 : **for** $i = 1$ à m **do**
3 : tirer $\psi_t^{[i]} \sim p(x_t | u_t, x_{t-1}^{[i]})$ ▷ Prediction
4 : $w_t^{[i]} = p(z_t, x_t^{[i]})$ ▷ Measurement update
5 : $\overline{\Psi_t} = \overline{\Psi_t} + \{x_t^{[i]}, w_t^{[i]}\}$
6 : **end for**
7 : **for** $i = 1$ à m **do**
8 : tirer i avec la probabilité $w_t^{[i]}$ ▷ Resampling
9 : ajouter $\psi_t^{[i]}$ à Ψ_t
10 : **end for**
Données en sortie : Ψ_t

Dans cet algorithme, l'étape de prédiction correspond à modifier les états représentés par toutes les particules selon le modèle d'évolution du système choisi. L'étape de *measurement update* détermine la pondération de chaque particule en fonction de la mesure et du modèle associé au capteur. L'étape de «ré-échantillonnage» (ou *resampling*) consiste alors à supprimer les particules présentant un poids faible et dupliquer celles dont la pondération est élevée. Différentes techniques existent pour effectuer cette étape de ré-échantillonnage [HSG06].

Resampling

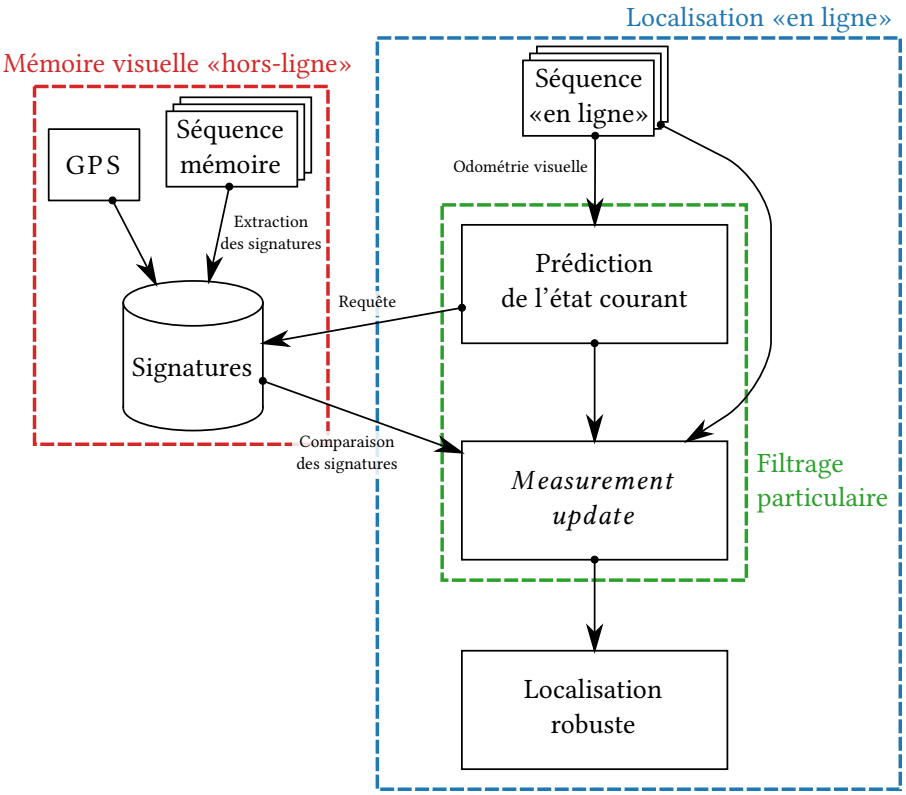
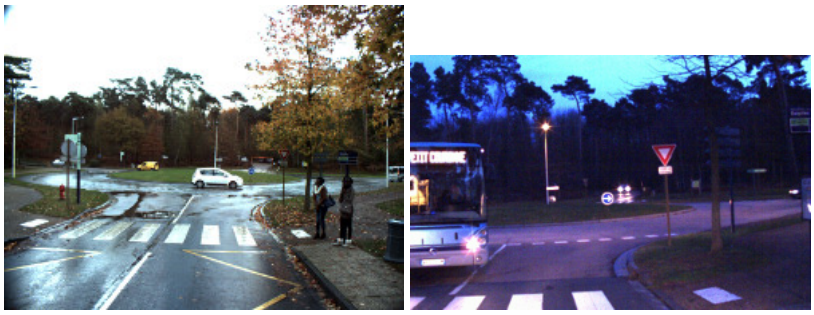


FIGURE 3.6 – Schéma global de l'approche mêlant PHROG, odométrie visuelle et filtrage particulaire.



(a) Séquence en mémoire (b) Séquence «en ligne»

FIGURE 3.7 – Exemples d'images tirées des séquences en mémoire et «en ligne».

Mise en pratique et résultats expérimentaux

La représentation schématique de l'approche utilisée dans cette partie est donnée en figure 3.6. Nous estimons ici l'évolution du système par odométrie visuelle *via* l'algorithme dit des «5 points» [Nis04]. La correction s'effectue grâce à l'approche PHROG présentée dans le deuxième chapitre de cette thèse. Nous utilisons ici notre propre ensemble de données. La première séquence a été enregistrée avec une caméra de résolution 1624 × 1234 pixels (figure 3.7b) et un GPS différentiel. La deuxième séquence a été faite en soirée plusieurs semaines plus tard avec une autre caméra (de résolution 752 × 480 pixels ; un exemple est donné en figure 3.7a). Sur la figure 3.8, nous donnons des positions des images en mémoire, la position estimée du véhicule en faisant la moyenne de la position des particules et un exemple de nuage de particules calculé au cours d'une étape de notre algorithme.

La figure 3.8 donne les positions moyennes (en vert) estimées à l'aide du filtre particulaire par rapport aux positions des images (en rouge) enregistrées précédemment avec la position GPS enregistrée au moment de l'acquisition. Nous remarquons que les positions estimées en ligne basées uniquement sur les capteurs de vision sont généralement proches de la position réelle sur route donnée par les données GPS de la carte. Parfois, certaines estimations successivement fausses de l'odométrie peuvent rendre le chemin estimé divergeant de la vérité terrain (comme on peut le constater au milieu du trajet effectué dans la figure 3.8) mais sont compensées grâce à une mise en correspondance cohérente avec l'approche PHROG.

3.3.3 Discussion

Nous avons proposé dans cette partie deux méthodes de localisation visuelle basée sur les descripteurs proposés dans le chapitre 2 de cette thèse. La première fait appel au

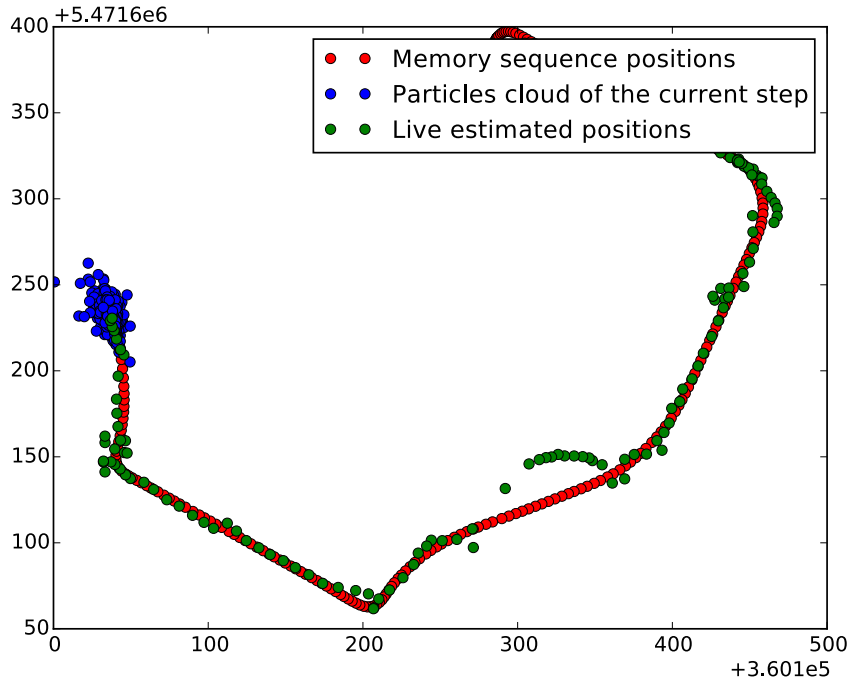


FIGURE 3.8 – Estimation de la localisation d'un système par odométrie visuelle, l'approche PHROG et un filtre particulaire.

descripteur global et un filtre Bayésien discret. Il permet d'envisager une localisation visuelle sans *a priori* et donc dans une problématique de type *robot kidnapping* telle que présentée en début de ce chapitre. Néanmoins, plus la mémoire sera composée de scènes nombreuses (et donc d'images à évaluer) plus le temps de calcul associé à cette méthode sera élevé. Une réflexion possible serait d'envisager une représentation des données en mémoire de manière structurée (en évaluant la dissemblance entre chaque scène et en ne conservant en mémoire que les images de scènes discriminantes par exemple) ou en améliorant la méthode d'association image afin de ne pas avoir à calculer la distance entre les deux signatures d'image à chaque test. Cette méthode peut s'envisager dans une problématique de localisation dans une carte topologique où les nœuds du graphe représentent les scènes observées.

La deuxième méthode s'approche plus d'une évaluation de la position du système dans une carte métrique à 2 dimensions (nous avons évalué le déplacement seulement dans le plan). Elle fait appel à une technique standard d'odométrie visuelle qui a elle-même ses défauts (en particulier lorsque l'on utilise des caméras monoculaires avec une

focale moyenne comme ce fut le cas pour nos tests). Ceci explique en partie la dérive cumulative engendrée cette technique, notamment si le système entreprend des virages serrés. Notre approche de description PHROG couplé au filtre particulaire permet de compenser en partie cette dérive afin de remettre l'estimation de la localisation plus proche de la vérité terrain. Cette méthode reste néanmoins loin d'être idéale et souffre de défauts si bien qu'une utilisation en l'état actuel requièrt l'usage d'autres capteurs et sources d'information pour une tâche de navigation autonome.

CONCLUSION ET PERSPECTIVES

Le sujet développé dans le cadre de cette thèse est celui de la localisation visuelle multimodale en robotique mobile. Dans un premier chapitre, nous avons présenté le contexte de la problématique abordée : celui de la robotique avec ses systèmes complexes, multi-capteurs, répondant aux interactions et contraintes nombreuses de l'environnement dans lequel ils sont plongés. Notre problématique se concentrant sur des solutions mettant en œuvre des capteurs de vision, nous avons également évoqué le processus de formation des images, de la projection physique des rayons lumineux réfléchis par les objets de la scène observée à leurs interactions avec le dispositif optique du système imageur et son capteur photoélectrique. Nous avons présenté quelques-uns des traitements en imagerie permettant de corriger et améliorer l'apparence générale des images. Nous avons ensuite dressé un aperçu des méthodes d'extraction de données utiles des images : descripteurs globaux proposant un distillat des informations utiles de l'image, méthodes d'échantillonnage des caractéristiques utiles (disposition arbitraire ou détection de points d'intérêt), descripteurs ponctuels, méthodes de compression ou représentations intermédiaires ainsi que la tendance actuelle à substituer certaines tâches de la chaîne de traitement par des réseaux de neurones profonds. Nous avons en outre évoqué les difficultés supplémentaires à prendre en compte lorsqu'il est nécessaire d'associer des informations visuelles provenant de capteurs de modalités différentes (en particulier dans le cas visible/infrarouge).

Au sein du deuxième chapitre, nous avons précisé la problématique de la localisation

visuelle multimodale à long terme, tâche finalement relativement proche des méthodes de recherche d'images par le contenu. Il s'agit en effet de disposer d'un premier jeu d'images correspondant à des scènes différentes dans une certaine modalité. Plusieurs semaines ou mois plus tard, de nouvelles acquisitions, éventuellement dans une autre modalité, doivent être associées avec ce jeu d'images antérieur. Nous avons ainsi proposé deux contributions : la première est un descripteur global prenant en compte les paramètres intrinsèques des caméras afin d'échantillonner les informations selon ce critère. Cette proposition fait l'objet d'une publication dans un article de conférence. Elle présente des résultats intéressants face aux méthodes de l'état de l'art (bien que plus coûteuse en temps de calcul) mais nécessite néanmoins une connaissance des paramètres de calibration des caméras ainsi que des points de vue tout de même proches entre les images utilisées en mémoire et les images nouvellement acquises. Afin de pallier cette dernière problématique, nous avons proposé une méthode d'extraction de caractéristiques ponctuelles. Nous avons dans un premier temps remarqué que les détecteurs de points n'avaient pas tous la même capacité à détecter les mêmes caractéristiques de l'environnement physique d'une modalité à l'autre. Nous avons ainsi mené une étude afin de déterminer quel détecteur avait la meilleure répétabilité face à la problématique de la multimodalité. Nous avons ensuite détaillé notre approche de description nommée PHROG (*Plural Histograms of Restricted Oriented Gradients*) qui fait l'objet d'une publication dans un article de conférence nationale ainsi que d'une publication dans un journal international. Nous avons mené une série d'expérimentations sur plusieurs bases d'images qui tendent à démontrer que notre proposition se démarque clairement des méthodes de l'état de l'art lorsqu'il s'agit d'associer des informations issues de capteurs de sensibilités spectrales éloignées (visible et infrarouge lointain). Il reste néanmoins des cas pour lesquels l'accumulation des contraintes (variations de l'environnement à long terme, changement de modalité, changement de résolution des images, changement de point de vue) provoque des résultats médiocres de notre proposition, bien que légèrement meilleurs que les solutions de l'état de l'art éprouvées.

Dans un troisième chapitre, nous avons évoqué la notion de cohérence temporelle ainsi que les avantages que l'on peut en tirer pour confirmer ou infirmer au cours du temps la réponse d'une méthode de recherche d'images par le contenu. Nous avons vu que le contexte se prête particulièrement bien à la mise en place d'une modélisation dans un cadre probabiliste Bayésien. Nous avons ainsi proposé l'application de deux méthodes de filtrage probabilistes : un filtre Bayésien discret permettant une localisation visuelle sans *a priori* lors de l'initialisation. Celle-ci approche modestement la problématique de la localisation visuelle dans une carte vaste sans *a priori* (problématique aussi connue sous le nom *robot kidnapping*) et pourrait être adaptée

à une localisation sur une carte topologique. La deuxième solution met en œuvre un filtre particulaire visant à effectuer un suivi du système sur une carte métrique 2D (dont les images sont géolocalisées) uniquement à l'aide d'une caméra monoculaire. Cette approche fait appel à une technique d'odométrie visuelle sujette à une potentielle dérive. Le filtre particulaire, couplé à l'approche PHROG présentée dans le deuxième chapitre de cette thèse permet de compenser partiellement cette dérive avec un temps de latence non négligeable. Ces derniers développements méritent toutefois des améliorations du modèle implémenté et une étude sur de nouvelles bases de tests plus approfondie. Les résultats en l'état actuel ne permettent pas d'envisager une précision de la localisation suffisante pour une navigation autonome et nécessiterait d'être couplée à d'autres techniques et capteurs.

La question de la navigation autonome à l'aide de techniques de vision par ordinateur demeure une problématique complexe. Au cours des travaux de cette thèse, nous avons abordé le problème sous l'angle de la localisation soumise aux contraintes de l'association de données robuste à long terme ainsi qu'au changement de modalité. De nombreuses pistes d'amélioration sont envisageables. On peut en effet approfondir l'étude des détecteurs de caractéristiques locales en prenant en compte des critères supplémentaires à la répétabilité. La phase de description peut également être améliorée, en modifiant le descripteur de manière à le rendre invariant aux rotations et changements d'échelle par exemple. De même, l'établissement d'une représentation intermédiaire peut être étudiée plus en profondeur : méthode de *clustering* choisie, méthode de quantification, méthode de *pooling* et métrique de comparaison. Enfin, les travaux menés sur l'implémentation de deux filtres probabilistes ne sont pas exhaustifs et mériteraient eux aussi une étude plus approfondie afin d'affiner les modèles mis en œuvre.

PUBLICATIONS

Les travaux menés au cours de cette thèse ont donné lieu aux publications suivantes :

Communications en conférences

— Bonardi Fabien, Samia Ainouz, Rémi Boutteau, Yohan Dupuis, Xavier Savatier, and Pascal Vasseur. «Localisation visuelle multimodale à long terme» *GRETSI 2017*.

— Bonardi Fabien, Samia Ainouz, Rémi Boutteau, Yohan Dupuis, Xavier Savatier, and Pascal Vasseur. «A Novel Global Image Description Approach for Long Term Vehicle Localization» *EUSIPCO 2017*.

Article de journal

— Bonardi Fabien, Samia Ainouz, Rémi Boutteau, Yohan Dupuis, Xavier Savatier, and Pascal Vasseur. «PHROG : A Multimodal Feature for Place Recognition.» *Sensors* 17, no. 5 (2017).

BIBLIOGRAPHIE

[AAS+16] Cristhian A Aguilera, Francisco J Aguilera, Angel D Sappa, Cristhian Aguilera, and Ricardo Toledo. Learning cross-spectral similarity measures with deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016.

[AST15] Cristhian A Aguilera, Angel D Sappa, and Ricardo Toledo. Lghd : A feature descriptor for matching across non-linear intensity variations. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 178–181. IEEE, 2015.

[AZ12] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2911–2918. IEEE, 2012.

[BETVG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3) :346–359, 2008.

[Bou10] Rémi Boutteau. *Reconstruction tridimensionnelle de l'environnement d'un robot mobile, à partir d'informations de vision omnidirectionnelle, pour la préparation d'interventions.* PhD thesis, Université de Rouen, 2010.

[BS11] Matthew Brown and Sabine Susstrunk. Multi-spectral sift for scene

category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 177–184. IEEE, 2011.

[CDM14] Guillaume Caron, Amaury Dame, and Eric Marchand. Direct model based visual tracking and pose estimation using mutual information. *Image and Vision Computing*, 32(1) :54–63, jan 2014.

[CLVZ11] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details : an evaluation of recent feature encoding methods. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2011.

[CRF11] Alexandre Chapoulie, Patrick Rives, and David Filliat. A spherical representation for efficient visual loop closing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pages 335–342. IEEE, 2011.

[Dam10] Amaury Dame. *A unified direct approach for visual servoing and visual tracking using mutual information.* PhD thesis, Rennes 1, 2010.

[DM10] Amaury Dame and E. Marchand. Une approche unifiée reposant sur l'information mutuelle pour l'asservissement visuel et le suivi différentiel. In *Proceedings of the 18e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, Caen, France, France, 2010.

[DM12] Amaury Dame and Eric Marchand. Second order optimization of mutual information for real-time image registration. *IEEE Transactions on Image Processing*, 21(9) :4190–4203, 2012.

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[FBS11] Damien Firmenichy, Matthew Brown, and S Susstrunk. Multispectral interest points for rgb-nir image registration. In *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP)*, pages 181–184. IEEE, 2011.

[HDF12] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 759–773. Springer, 2012.

[HL07] Gerald C Holst and Terrence S Lomheim. *CMOS/CCD sensors and camera systems*, volume 408. JCD publishing USA, 2007.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[HSG06] Jeroen D Hol, Thomas B Schon, and Fredrik Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[LBKS13] H. Lategahn, J. Beck, B. Kitt, and C. Stiller. How to learn an illumination robust image feature for place recognition. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, Australia, 2013.

[LK+81] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[LL09] Hyon Lim and Young Sam Lee. Real-time single camera slam using fiducial markers. In *ICCAS-SICE, 2009*, pages 177–182. IEEE, 2009.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110, 2004.

[MA13] Tarek Mouats and Nabil Aouf. Multimodal stereo correspondence based on phase congruency and edge histogram descriptor. In *Proceedings of the 16th International Conference on Information Fusion (FUSION)*, pages 1981–1987. IEEE, 2013.

[MB13] Marina Magnabosco and Toby P Breckon. Cross-spectral visual simultaneous localization and mapping (slam) with sensor handover. *Robotics and Autonomous Systems*, 61(2) :195–208, 2013.

[Mor80] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1980.

[MPM15] Dmytro Mishkin, Michal Perdoch, and Jiri Matas. Place recognition with wxbs retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Visual Place Recognition in Changing Environments*, 2015.

[MS05] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10) :1615–1630, 2005.

[MTS+05] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2) :43–72, 2005.

[MV12] William Maddern and Stephen Vidas. Towards robust night and day place recognition using visible and thermal imaging. *RSS 2012 : Beyond laser and vision : Alternative sensing techniques for robotic perception*, 2012.

[Nis04] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(6) :756–770, 2004.

[NSBS14] Tayyab Naseer, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. Robust visual robot localization across seasons using network flows. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. AAAI Press, 2014.

[NSP13] Peer Neubert, Niko Sunderhauf, and Peter Protzel. Appearance change prediction for long-term navigation across seasons. In *Proceedings of the European Conference on Mobile Robots (ECMR)*, pages 198–203. IEEE, 2013.

[OT01] Aude Oliva and Antonio Torralba. Modeling the shape of the scene : A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3) :145–175, 2001.

[PCI+08] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization : Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[PSM10] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 143–156. Springer, 2010.

[RCAC+14] Pablo Ricaurte, Carmen Chilán, Cristhian A Aguilera-Carrasco, Boris X Vintimilla, and Angel D Sappa. Feature point descriptors : Infrared and visible spectra. *Sensors*, 14(2) :3690–3701, 2014.

[SB13] Aleksandra A Sima and Simon J Buckley. Optimizing sift for matching of short wave infrared and visible wavelength images. *Remote Sensing*, 5(5) :2037–2056, 2013.

[Sin01] Amit Singhal. Modern information retrieval : A brief overview. *IEEE Data Eng. Bull.*, 24(4) :35–43, 2001.

[SNP13] Niko Sünderhauf, Peer Neubert, and Peter Protzel. Are we there yet ? challenging seqslam on a 3000 km journey across all four seasons. In *Pro-*

ceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Long-Term Autonomy, page 2013, 2013.

[ST94] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600. IEEE, 1994.

[SZ03] Josef Sivic and Andrewweyand Zisserman. Video google : A text retrieval approach to object matching in videos. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV)*, pages 1470–1477. IEEE, 2003.

[Sze10] Richard Szeliski. *Computer vision : algorithms and applications*. Springer Science & Business Media, 2010.

[TBF+05] Sebastian Thrun, Wolfram Burgard, Dieter Fox, et al. *Probabilistic robotics*, volume 1. MIT press Cambridge, 2005.

[TM08] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors : a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3) :177–280, 2008.

[vGGVS08] Jan C van Gemert, Jan-Mark Geusebroek, Cor J Veenman, and Arnold WM Smeulders. Kernel codebooks for scene categorization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 696–709. Springer, 2008.

[WE14] Ryan W Wolcott and Ryan M Eustice. Visual localization within lidar maps for automated urban driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 176–183. IEEE, 2014.

[WKP16] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. *arXiv preprint :1602.05314*, 2016.

[WYY+10] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3360–3367. IEEE, 2010.

[ZWT99] Zhiqiang Zheng, Han Wang, and Eam Khwang Teoh. Analysis of gray level corner detection. *Pattern Recognition Letters*, 20(2) :149–162, 1999.

[ZYZH10] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S Huang. Image classification using super-vector coding of local image descriptors. In *Proceedings*

of the European Conference on Computer Vision (ECCV), pages 141–154. Springer, 2010.

Résumé

L'expression navigation autonome désigne les méthodes visant à automatiser les déplacements d'un robot. Les travaux présentés se concentrent sur la problématique de la localisation en milieu extérieur et approchent la problématique de la localisation visuelle soumise à la fois à un changement de capteurs vision (géométrie et modalité) ainsi qu'aux changements de l'environnement à long terme, contraintes combinées encore très peu étudiées dans l'état de l'art. La contribution majeure de cette thèse porte sur la phase de description et compression des données issues des images sous la forme d'un histogramme de mots visuels, nommée PHROG. Les résultats obtenus sur plusieurs bases d'images démontrent une amélioration des performances de reconnaissance de scènes comparés aux méthodes de l'état de l'art. La nature séquentielle des images acquises dans un contexte de navigation est prise en compte par la suite afin de filtrer et supprimer des estimations de localisation aberrantes. Deux applications de filtrage probabiliste sont proposés : une première solution définit un modèle de déplacement simple du robot avec un filtre d'histogrammes et la deuxième met en place un modèle plus évolué faisant appel à l'odométrie visuelle au sein d'un filtre particulaire.

Mots-clefs : Vision multimodale visible/infrarouge, indexation d'image, localisation visuelle, navigation autonome, filtrage probabiliste

Abstract

Autonomous navigation field gathers the set of algorithms which automate the moves of a mobile robot. The case study of this thesis focuses on the outdoor localisation issue with additionnal constraints : the use of visual sensors only with variable specifications (geometry, modality, *etc*) and long-term apparence changes of the surrounding environment. Both types of constraints are still rarely studied in the state of the art. Our main contribution concerns the description and compression steps of the data extracted from images. We developped a method called PHROG which represents data as a visual-words histogram. Obtained results on several images datasets show an improvment of the scenes recognition performance compared to methods from the state of the art. In a context of navigation, acquired images are sequential such that we can envision a filtering method to avoid faulty localisation estimation. Two probabilistic filtering approaches are proposed : a first one defines a simple movement model with a histograms filter and a second one sets up a more complex model using visual odometry and a particules filter.

Keywords : Multimodal visible/infrared vision, image retrieval, visual localisation, autonomous navigation, probabilistic filtering