BetaSAC et OABSAC, deux nouveaux 'echantillonnages conditionnels pour RANSAC

Antoine Méler

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques Appliquées et Informatique**

Arrêté ministérial :

Présentée par

Antoine Méler

Thèse dirigée par **James L. Crowley**

préparée au sein **du Laboratoire d'Informatique de Grenoble**
et de **l'école doctorale de Mathématiques, Sciences et Technologies de l'Information**

BetaSAC et OABSAC, deux nouveaux échantillonnages conditionnels pour RANSAC

Thèse soutenue publiquement le ,
devant le jury composé de :

Valérie Perrier
Professeur à Grenoble INP, Présidente
Adrien Bartoli
Professeur à l'université d'Auvergne - ISIT, Rapporteur
Julie Delon
LTCI, CNRS, Télécom ParisTech, Rapporteur
Patrick Pérez
Technicolor, Examinateur
James L. Crowley
Professeur, Grenoble INP, Directeur de thèse

Résumé

L'algorithme RANSAC (*Random Sample Consensus*) [19] est l'approche la plus commune pour l'estimation robuste des paramètres d'un modèle en vision par ordinateur. C'est principalement sa capacité à traiter des données contenant potentiellement plus d'erreurs que d'information utile qui fait son succès dans ce domaine où les capteurs fournissent une information très riche mais très difficilement exploitable. Depuis sa création, il y a trente ans, de nombreuses modifications ont été proposées pour améliorer sa vitesse, sa précision ou sa robustesse.

Dans ce travail, nous proposons d'accélérer la résolution d'un problème par RANSAC en utilisant plus d'information que les approches habituelles. Cette information, calculée à partir des données elles-même ou provenant de sources complémentaires de tous types, nous permet d'aider RANSAC à générer des hypothèses plus pertinentes.

Pour ce faire, nous proposons de distinguer quatre degrés de qualité d'une hypothèse : la *non contamination*, la *cohésion*, la *cohérence* et enfin la *pertinence*. Puis nous montrons à quel point une hypothèse non contaminée par des données erronées est loin d'être pertinente dans le cas général. Dès lors, nous nous attachons à concevoir un algorithme original qui, contrairement aux méthodes de l'état de l'art, se focalise sur la génération d'échantillons *pertinents* plutôt que simplement *non contaminés*.

Notre approche consiste à commencer par proposer un modèle probabiliste unifiant l'ensemble des méthodes de réordonnancement de l'échantillonnage de RANSAC. Ces méthodes assurent un guidage du tirage aléatoire des données tout en se prémunissant d'une mise en échec de RANSAC. Puis, nous proposons notre propre algorithme d'ordonnancement, BetaSAC, basé sur des tris conditionnels partiels. Nous montrons que la conditionnalité du tri permet de satisfaire des contraintes de cohérence des échantillons formés, menant à une génération d'échantillons pertinents dans les premières itérations de RANSAC, et donc à une résolution rapide du problème. L'utilisation de tris partiels plutôt qu'exhaustifs, quant à lui, assure la rapidité et la randomisation, indispensable à ce type de méthodes.

Dans un second temps, nous proposons une version optimale de notre méthode, que l'on appelle OABSAC (pour Optimal and Adaptative BetaSAC), faisant intervenir une phase d'apprentissage hors ligne. Cet apprentissage a pour but de mesurer les propriétés caractéristiques du problème spécifique que l'on souhaite résoudre, de façon à établir automatiquement le paramétrage optimal de notre algorithme. Ce paramétrage est celui qui doit mener à une estimation suffisamment précise des paramètres du modèle recherché en un temps (en secondes) le plus court.

Les deux méthodes proposées sont des solutions très générales qui permettent d'intégrer dans RANSAC tout type d'information complémentaire utile à la résolution du problème. Nous montrons l'avantage de ces méthodes pour le problème de l'estimation d'homographies et de géométries épipolaires entre deux photographies d'une même scène. Les gains en vitesse de résolution du problème peuvent atteindre un facteur cent par rapport à l'algorithme RANSAC classique.

Abstract

RANSAC algorithm (*Random Sample Consensus*) [19] is the most common approach for the problem of robust parameters estimation of a model in computer vision. This is mainly its ability to handle data containing more errors than potentially useful information that made its success in this area where sensors provide a very rich but noisy information. Since its creation thirty years ago, many modifications have been proposed to improve its speed, accuracy and robustness.

In this work, we propose to accelerate the resolution of a problem by using RANSAC with more information than traditional approaches. This information, extracted from the data itself or from complementary sources of all types, is used help generating more relevant RANSAC hypotheses.

To do this, we propose to distinguish four degrees of quality of a hypothesis : inlier, consistent, coherent or suitable sample. Then we show how an inlier sample is far from being relevant in the general case. Therefore, we strive to design a novel algorithm which, unlike previous methods, focuses on the generation of suitable samples rather than inlier ones.

We begin by proposing a probabilistic model unifying all the RANSAC reordered sampling methods. These methods provide a guidance of the random data selection without impairing the search. Then, we propose our own scheduling algorithm, BetaSAC, based on conditional partial sorting. We show that the conditionality of the sort can satisfy consistency constraints, leading to a generation of suitable samples in the first iterations of RANSAC, and thus a rapid resolution of the problem. The use of partial rather than exhaustive sorting ensures rapidity and randomization, essential to this type of methods.

In a second step, we propose an optimal version of our method, called OABSAC (for Optimal and Adaptive BetaSAC), involving an offline learning phase. This learning is designed to measure the properties of the specific problem that we want to solve in order to determine automatically the optimal setting of our algorithm. This setting is the one that should lead to a reasonably accurate estimate of the model parameters in a shortest time (in seconds).

The two proposed methods are very general solutions that integrate into RANSAC any additional useful information. We show the advantage of these methods to the problem of estimating homographies and epipolar geometry between two photos of the same scene. The speed gain compared to the classical RANSAC algorithm can reach a factor of hundred.

Notations

Général

$\mathscr{D}$	Ensemble des données
$\mathscr{D}_0$	Données de $\mathscr{D}$ considérées comme erronées d'après la vérité terrain
$\mathscr{D}_1$	Données de $\mathscr{D}$ considérées comme correctes d'après la vérité terrain
N	Nombre de données dans $\mathscr{D}$
d	Une données
D	Variable aléatoire dans l'ensemble des données
d_i	i ième donnée
$d_{(i)}$	Donnée de rang i après tri
$d_{(i)_s}$	Donnée de rang i après tri sachant s
s	Échantillon de données (*sample*)
S	Variable aléatoire dans l'ensemble des échantillons
m	Taille minimale d'un échantillon pour pouvoir instancier un modèle.
t	Compteur d'itérations de RANSAC
I	Taux de données correctes (non aberrantes)
P_I	Probabilité de sélectionner un échantillon complet (de taille m) non contaminé (i.e. sans donnée erronée)
$\mathbb{E}[X]$	Espérance de la variable aléatoire X

Correspondances entre points d'intérêts

$P(d), P'(d)$	Les deux points d'intérêt formant la correspondance d
$Aff(d)$	Matrice affine associée à la correspondance d
$Sim(d)$	Matrice de similitude associée à la correspondance d
$Match(d)$	Confiance de la mise en correspondance d.

BetaSAC & OABSAC

T	Nombre d'itérations guidées
l	Taille de l'échantillon partiel, en cours de formation. $l = \|s\|, l \leq m$
n	Nombre de données sélectionnées pour n'en garder qu'une
$\vec{n}$	Vecteur du nombre de données sélectionnées pour chaque taille l
i_l	Rang du point à conserver parmi les n points sélectionnés, en fonction de l
$\vec{i}(t)$	Vecteur des valeurs $i_l, \forall l \in \{0, \cdots, m-1\}$. Nous l'appelons le *vecteur de sélection* pour l'itération t
$B_{i/n}$	Variable aléatoire de BetaSAC dans l'espace des données
$B_{\vec{i}/n}$	Variable aléatoire de BetaSAC dans l'espace des échantillons de taille m
$t \mapsto \vec{i}(t)$	Nous appelons cette fonction la *Stratégie*
$\mathscr{M}_k$	*Mesure* (i.e. critère de qualité d'un échantillon complet ou non) numéro k (voir le tableau 5.3)
t_s	Temps de la création d'un échantillon complet (de taille m)
$t_{mes}(l)$	Temps mis pour effectuer l'ensemble des mesures sur un échantillon de taille l
$t_{k^{th}}(n)$	Durée de la sélection du k-ième élément parmi n
t_t	Durée du test d'une hypothèse sur une donnée

Table des matières

1

Introduction

Sommaire

1.1 Problématique

La problématique que l'on cherche à résoudre s'inscrit dans le contexte de la vision par ordinateur et plus précisément la détection d'objets dans les images. Nous commençons par présenter succintement ce domaine des mathématiques appliquées avant de présenter le problème de l'estimation robuste de la pose d'un objet dans une image.

1.1.1 La détection d'objets dans les images

La détection d'objets dans les images et les vidéos est l'un des grands problèmes de la vision par ordinateur. C'est un exercice difficile en raison de la grande variabilité de l'apparence en fonction de la pose, de l'éclairage, des occultations, des autres objets présents etc.

On considère souvent le système de reconnaissance d'objets mis au point par Lawrence G. Roberts en 1965 comme le début de la vision par ordinateur. Son programme était capable de déduire de l'image des contours projetés (obtenue à partir de techniques de traitement de l'image) d'un solide composé de blocs simples, une reconstruction 3D. S'en est suivi d'autres réalisations de systèmes de reconnaissance à partir de modèles 3D de l'objet. Ces méthodes se focalisent sur l'utilisation de la géométrie de l'objet pour en déduire leur variation d'apparence avec un changement de point de vue ou d'illumination. Les contours de l'objet pouvant être prédits précisément sous n'importe quelle projection, ces méthodes s'attachent alors à détecter des primitives géométriques simples dans les photos, susceptibles de correspondre aux bords de l'objet [38]. Les plus importantes d'entre elles sont les lignes et les cercles. Cependant, ces primitives sont très sensibles aux variations de point de vue et ne sont que rarement liées à la géométrie de l'objet lorsque celui-ci est texturé.

Alors dans les années 90, le domaine a connu un tournant avec l'émergence de nouvelles méthodes travaillant à partir de vues 2D de l'objet à reconnaître. Une des méthodes emblématiques est "eigenface"

(1991) [50, 60]. Il s'agit du premier système de reconnaissance de visage efficace et robuste. L'idée de cette approche est de calculer les vecteurs propres d'un ensemble de vecteurs dont chacun représente une image de visage en niveau de gris. Chaque image de l'ensemble d'apprentissage est une capture du visage dans des conditions d'illumination, une expression et une orientation particulières. Ainsi, les quelques vecteurs propres les plus significatifs capturent presque toute la variabilité d'apparence du visage avec l'illumination. Cette même idée a ensuite été utilisée pour reconnaître des objets génériques sous différentes illuminations [3] mais aussi différents points de vue [39]. Plus tard, les chercheurs se sont attelé à remplacer les modèles descriptifs par des modèles discriminants, car en effet, le but est de distinguer un objet parmi d'autres. Pour cela, de nombreux classifieurs différents ont été utilisés : la méthode des k plus proches voisins, des réseaux de neurones, l'analyse discriminante linéaire, la machine à vecteurs de support (SVM) ou encore le boosting [51, 26, 2, 46, 49]. Ces méthodes ont montré de très bons résultats en reconnaissance d'objets sous variation de point de vue et d'illumination. Par contre, elles ont toutes le défaut d'être sensibles aux occultations.

Pour gérer les occultations il a fallu changer complètement de paradigme. Les objets n'ont plus été représentés par des vecteurs calculés sur l'image entière mais par un ensemble de caractéristiques locales d'apparence. Le plus souvent, les caractéristiques utilisées sont les points d'intérêt, apparaissant sur des singularités de la fonction d'intensité ayant la propriété d'être invariantes par changement illumination, translation et rotation. Le plus célèbre d'entre eux est certainement le détecteur de Harris [23], mais il en existe de nombreux, plus ou moins stables, rapides à calculer et précis [31, 22, 32]. De nombreux travaux on également donné à ces caractéristiques l'invariance par changement d'échelle, notamment la thèse de T. Lindeberg [28]. Cette invariance est le plus souvent obtenue par projection de l'image sur une base de fonctions gaussiennes et leurs dérivées, qui sont une bonne approximation des champs récepteur des neurones du systèmes visuel humain [28, 16]. L'une des plus grandes avancées dans ce domaine est réalisée par D. Lowe en 1999 avec son descripteur SIFT (Scale-Invariant Feature Transform) [30, 31]. Enfin, des efforts récents ont permis d'obtenir l'invariance affine [36, 59, 61, 25]. La transformation affine n'est cependant qu'une approximation des distorsions dues à la projection d'un objet dans une image, qui sont en fait de type homographique. Mais plus une caractéristique est locale, plus cette approximation est suffisante, ce qui est leur avantage principal. Encore une fois, les progrès les plus récents ont été obtenus grâce à l'utilisation de méthodes d'apprentissage aussi bien pour la détection de caractéristiques stables [48, 58] que pour leur description invariante [6, 24].

1.1.2 Estimation robuste de la pose d'un objet dans une image

Nous abordons le problème de la détection d'objets rigides par une approche par caractéristiques locales multi-échelles. C'est-à-dire que l'information de l'image, que l'on peut considérer dense, est rendue éparse en ne conservant qu'un ensemble de caractéristiques locales stables. La stabilité de ces caractéristiques est la propriété de les retrouver sur d'autres images de la même scène, ou du même objet, malgré des variations de point de vue, d'angle ou encore de luminosité.

Une fois les caractéristiques détectées sur deux images, on les met en correspondance en fonction de leur probabilité de localiser un même point de l'objet ou de la scène. On obtient alors un ensemble de correspondances, souvent fortement erroné. Des exemples de résultats sont donnés dans la figure 1.1.

L'hypothèse de rigidité de la scène ou de l'objet implique que le sous-ensemble de correspondances liant effectivement un même point de la scène, projeté sur les deux images, peut être décrit par un modèle homographique ou épipolaire. Le problème à résoudre est alors celui de l'estimation des paramètres d'un modèle connu à partir de données, seulement en partie décrites par celui-ci, l'autre partie étant des erreurs pour lesquelles on ne connaît aucun modèle. Le travail effectué dans cette thèse est alors de bâtir un algorithme rapide d'estimation robuste des paramètres de ce modèle.

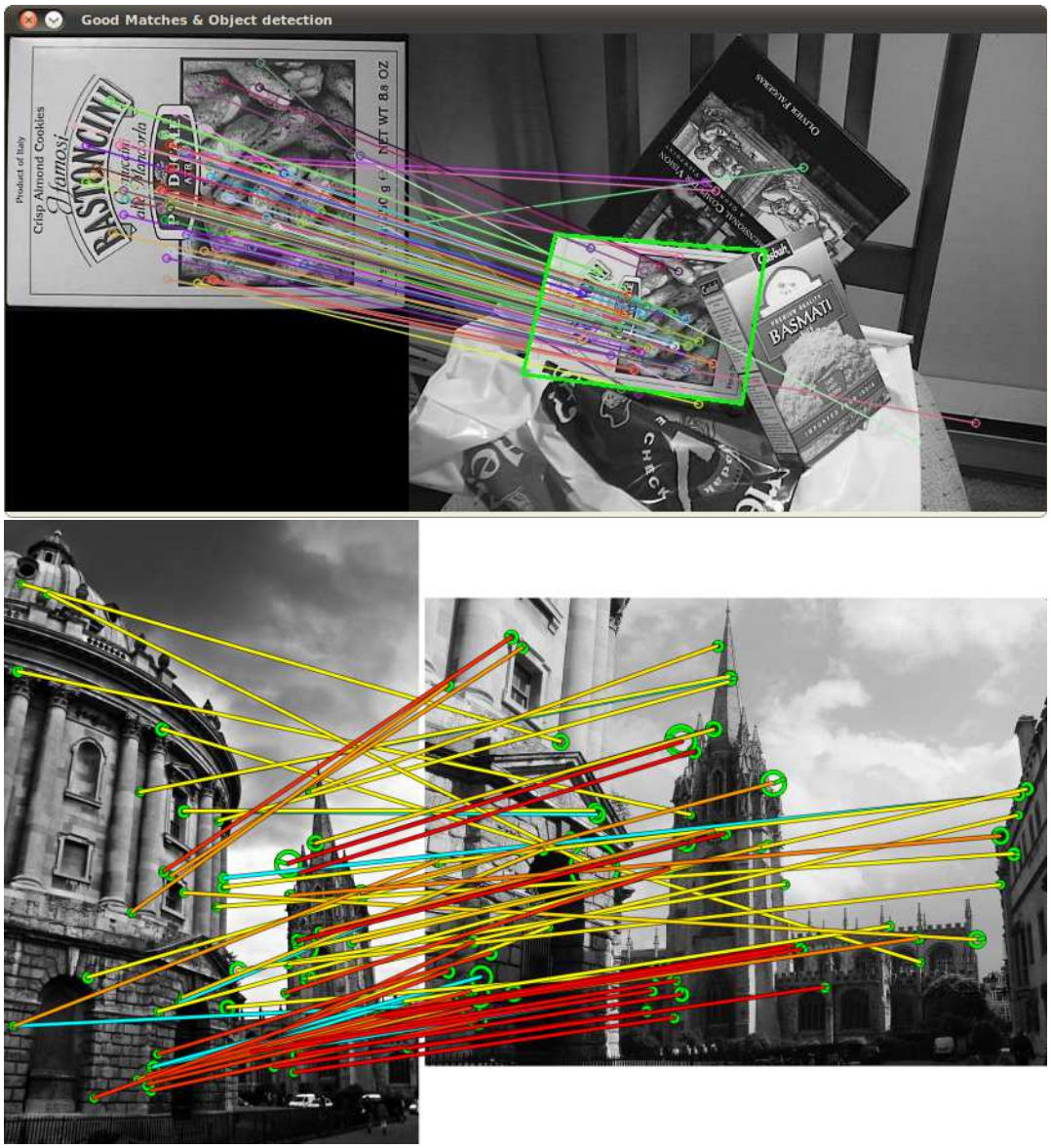


FIGURE 1.1 – Deux exemples de mises en correspondances de caractéristiques locales d'apparence issus de [5] et [54]. On constate que toutes les correspondances ne sont pas exactes. Ce type de résultats est la donnée d'entrée de notre problème.

1.2 Notre Approche

1.2.1 Réalisation

Notre travail est fondé sur l'algorithme d'estimation robuste RANSAC (RANdom SAmple Consensus). RANSAC est un algorithme itératif non-déterministe nécessitant, dans les cas les plus difficiles, un très grand nombre d'itérations pour avoir une confiance suffisante en le résultat. Nous proposons de réduire considérablement le nombre d'itérations moyen de RANSAC en le rendant capable d'intégrer plus d'information que dans sa description initiale par Fischler et Bolles en 1981.

A chacune de ses itérations, RANSAC commence par sélectionner aléatoirement un petit ensemble de données. Si toutes les données sélectionnées sont correctes, alors l'itération a une chance de fournir une bonne approximation des paramètres recherchés. Si, au contraire, une ou plusieurs données sont erronées, alors l'itération ne permettra pas d'obtenir une estimation des paramètres. Le travail présenté dans cette thèse a abouti à deux méthodes, BetaSAC et OABSAC, proposant une nouvelle sélection aléatoire.

1.2.1.1 BetaSAC

BetaSAC est la première modification de RANSAC que nous proposons. Son nom est issu de la loi de probabilité bêta car la sélection aléatoire uniforme de RANSAC est remplacée par une sélection non uniforme obéissant à la loi bêta (équation 1.1, où Beta désigne la fonction bêta).

$$f(x;\alpha,\beta) = \frac{1}{\mathrm{Beta}(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}\ \mathbb{1}_{[0,1]}(x), \qquad \mathrm{Beta}(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\,dt \tag{1.1}$$

Fonction de densité de la loi bêta

La sélection n'étant plus uniforme, on peut la guider en utilisant de l'information supplémentaire, de façon à sélectionner plus souvent des groupes de données correctes. Ainsi, les paramètres du modèle recherché peuvent être estimés plus rapidement.

1.2.1.2 OABSAC

OABSAC (Optimal and Adaptative BetaSAC) est une version optimale de notre algorithme BetaSAC, en terme de temps nécessaire pour résoudre un problème d'estimation. Cette version contient plus de paramètres que BetaSAC, mais elle fait intervenir une phase d'apprentissage qui les définit au mieux, de façon automatique. En outre, certains de ces paramètres peuvent évoluer au cours des itérations de façon à s'adapter en ligne aux données traitées.

1.2.2 Résultats

Nous avons évalué notre méthode sur le problème de la détection d'objets plans puis tridimensionnels à l'aide d'un détecteur de points d'intérêt invariant à une transformation affine. Nos résultats nous ont montré des accélérations considérables (jusqu'à un facteur cent) par rapport à la méthode RANSAC classique et aux méthodes de l'état de l'art. La figure 1.2 est un exemple de résultat obtenu.

1.3 Apports

La méthode mise au point dans ce travail permet d'accélérer considérablement la phase d'estimation de la transformation géométrique, commune à de nombreuses applications en vision par ordinateur telles que la détection d'objets rigides, la création de panoramas, la reconstruction 3D d'une scène, la cartographie et localisation simultanées etc. Or, cette phase est très souvent la plus coûteuse en temps parmi

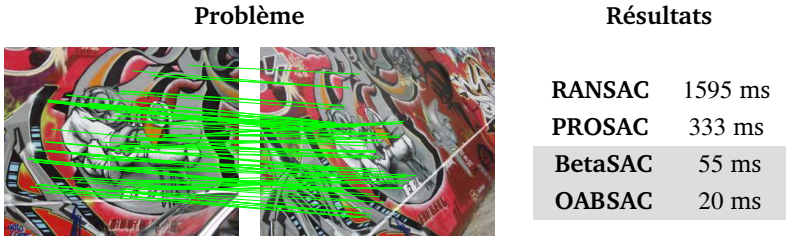


FIGURE 1.2 – Exemple de détection d'un objet plan

toutes les étapes du pipeline de traitement des images. Notre travail est donc susceptible non seulement d'apporter un gain en vitesse significatif à la plupart de ces applications mais de rendre accessible des nouveaux cas d'application, jugés trop coûteux jusque là. En pense notamment aux environnement visuels difficiles tels que les scènes urbaines, qui souffrent de fortes répétitions de motifs ou les scènes d'intérieurs qui cumulent bien souvent les répétitions avec une pauvreté de l'information.

1.4 Organisation du manuscrit

La suite de ce manuscrit comporte six parties. Nous commençons par montrer comment une approche par caractéristiques locales d'apparence permet de reformuler le problème de la détection d'objets dans les images en un problème d'estimation robuste des paramètres d'un modèle mathématique. En effet, si l'image de référence de l'objet (appelée image d'apprentissage) et l'image de test sont toutes deux traitées avec le même détecteur de caractéristiques locales (nous utilisons les points d'intérêt), alors le problème est de trouver la cohérence entre les deux ensembles de caractéristiques. Dans notre travail, l'objet étant considéré comme rigide, cette cohérence est une transformation géométrique de type homographique (si l'objet est plan ou bien si le point de vue des deux images est le même) ou épipolaire (dans le cas général). Une fois notre problème reformulé en estimation robuste des paramètres d'un modèle mathématique, nous présentons ce nouveau problème plus général et nous décrivons ses trois grands paradigmes de résolution que sont la transformée généralisée de Hough, la méthode RUDR (Recognition Using Decomposition and Randomization) et la méthode RANSAC (RANdom SAmple Consensus).

Notre travail s'inscrit parfaitement dans le paradigme de RANSAC, qui est de loin le plus utilisé dans notre domaine d'application. La partie suivante est alors une description plus détaillée de cette approche, dont nous distinguons trois grandes parties : la création d'une hypothèse à partir d'une sous-ensemble des données (échantillonnage), sa vérification et l'évaluation d'un test d'arrêt. Puis nous présentons quelques unes des nombreuses améliorations proposées ces dernières années. Chacune des améliorations présentées s'attache à améliorer la vitesse en travaillant en amont (sur l'échantillonnage) ou en aval (sur la vérification), la précision du résultat ou bien la robustesse.

Pour qu'une méthode de type RANSAC parvienne à estimer correctement un modèle, il faut que l'échantillon utilisé pour en inférer les paramètres soit de bonne qualité. Or, cette notion de qualité n'a pas été suffisamment bien définie par les travaux précédents. Alors, dans la section suivante, nous commençons par proposer une nomenclature des différents degrés de qualité d'un échantillon en proposant quatre propriétés que nous définissons : la *non contamination*, la *cohésion*, la *cohérence* et enfin la *pertinence*. Une hypothèse non contaminée par des données erronées est loin d'être pertinente dans le cas général. Alors, nous nous attacherons à concevoir un algorithme original qui, contrairement aux méthodes de l'état de l'art, se focalise sur la génération d'échantillons *pertinents* plutôt que simplement *non contaminés*. Puis, nous remarquons que les améliorations dites par échantillonnage réordonné présentent des propriétés particulièrement intéressantes qui peuvent être unifiées dans un formalisme commun, que nous bâtissons, avant de proposer un réordonnancement plus général, capable en théorie de former des échantillons de qualité maximale. La propriété fondamentale, et originale, de ce nouvel échantillonnage est de générer un échantillon en ne sélectionnant pas les éléments de façon

indépendante. Un échantillon ainsi généré peut alors posséder des propriétés de cohérence et donc une qualité inédites.

Nous décrivons précisément la variable aléatoire d'échantillonnage que nous proposons d'utiliser pour générer de façon efficace des échantillons-hypothèse de qualité dans la partie suivante. Cette variable est dérivée de la loi de probabilité Beta. La loi Beta est la *statistique d'ordre* de la loi uniforme. C'est-à-dire que c'est la loi de probabilité de la i-ième plus petite valeur parmi n valeurs générées par une loi uniforme. Nous montrons comment on peut alors sélectionner des valeurs dans un tri quelconque à partir d'un rang généré avec la loi Beta. Pour ce faire, il n'y a pas à effectuer de tri complet des données. En contrepartie la loi du rang de la donnée sélectionnée possède une un écart-type non nul, ce qui n'est pas un problème puisque notre sélection se doit de comporter une part d'aléatoire. Cette capacité à sélectionner des données en maitrisant en partie leur rang dans un tri nous permet de former des échantillons pertinents à partir d'un critère de tri reflétant la cohérence des données entre elles. Différents critères sont proposés et combinés pour le cas d'application qui nous intéresse. Notre variable aléatoire de sélection des échantillons est utilisée dans deux algorithmes différents. Le premier, que nous nommons BetaSAC, n'est pas optimale mais particulièrement simple à réaliser. Le second, appelée OABSAC (pour Optimal and Adaptative BetaSAC), est une variante optimale de BetaSAC, nécessitant une phase d'apprentissage avant d'être opérationnel. Nous décrivons ces deux algorithmes et montrons qu'ils satisfont les propriétés qui nous intéressent.

Enfin, nous réalisons un ensemble d'expériences sur le problème de la détection d'objets plans puis tridimensionnels à l'aide d'un détecteur de points d'intérêt invariant à une transformation affine.

2

Problème de l'estimation robuste des paramètres d'un modèle et méthodes de résolution

Sommaire

2.1 Problème de l'estimation robuste des paramètres d'un modèle

La détection d'objets rigides peut être abordé comme un problème, plus général, d'estimation robuste des paramètres d'un modèle connu, à partir d'un ensemble de données dont une partie est due au modèle recherché. Le problème de l'estimation robuste des paramètres d'un modèle à partir de données en partie erronées est un vieux problème des mathématiques appliquées. Aujourd'hui, il est devenue une étape cruciale de nombreuses applications en vision par ordinateur. Pourtant, il est encore loin d'être définitivement résolu. Nous proposons dans cette section une rapide présentation du problème.

2.1.1 Entrée du problème

2.1.1.1 Données

L'entrée du problème est un ensemble de données dont une partie est générée par un phénomène pour lequel on connaît un modèle. Ces données dont on connaît le modèle sont néanmoins sujettes à un *bruit de mesure* qui fait qu'elles coïncident avec le modèle à une certaine marge d'erreur près. Cette marge d'erreur peux faire partie des entrées du problème, ou bien être déterminée automatiquement par la méthode de résolution. La figure 2.1 est un exemple d'entrée dans le cas où l'on recherche une droite (un alignement de points) dans un nuage de points en deux dimensions.

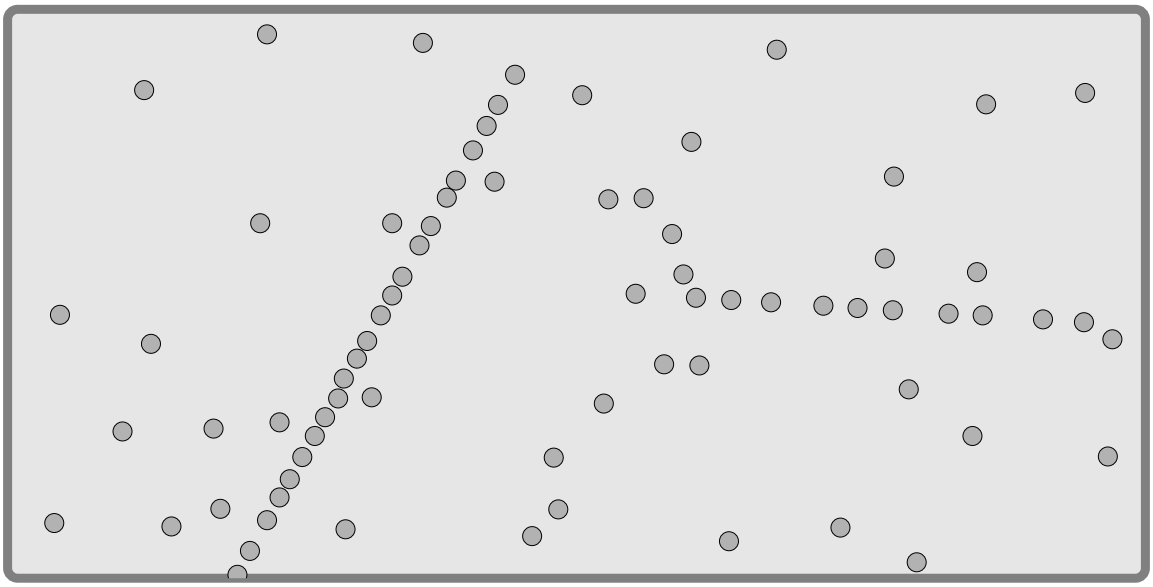


FIGURE 2.1 – Exemple d'entrée : nuage de points en deux dimensions dont une partie des points est générée par un modèle de droite. Dans cet exemple, on peut considérer qu'il existe deux instances du modèle de droite dans les données, car deux groupes de points alignés "ressortent".

2.1.1.2 Modèle

Dans ce travail, nous ne nous intéressons pas au problème de la détermination automatique du bon modèle, alors il doit être fourni en entrée. Dans le cas de la recherche d'une droite dans un nuage de points en deux dimensions (x,y), on peut par exemple utiliser un de ces deux modèles équivalents :

- $\mathbf{a} \cdot x + \mathbf{b} \cdot y + 1 = 0$
- $-\left(\frac{cos(\theta)}{\mathbf{r}}\right) \cdot x - \left(\frac{sin(\theta)}{\mathbf{r}}\right) \cdot y + 1 = 0$

2.1.2 Sortie du problème

La sortie du programme est l'ensemble des paramètres du modèle présent dans les données. Dans le cas ou plusieurs instances du modèle sont présentes dans les données, le programme doit les trouver puis retirer des données, les une après les autres.

Il est évident que le problème de l'estimation robuste des paramètres d'un modèle n'est en générale pas parfaitement bien posé. En effet, une partie des données erronées peuvent sembler avoir été générées par le modèle recherché si la marge d'erreur de mesure que l'on admet est trop élevée. A l'inverse, des données effectivement générées par le modèle recherché peuvent apparaître comme des erreurs si la marge est trop restrictive ou si les paramètres du modèle sont estimés de manière trop imprécises. De plus, à partir de combien de données correspondant au modèle recherché peut-on conclure que l'on a trouvé une instance du modèle ? Dans notre exemple, toute droite passant par un des points pourrait être considérée comme une instance du modèle représentée dans les données. Pour toutes ces raisons, il n'existe pas, dans le cas général, de critère communément utilisé capable d'affirmer si l'ensemble des paramètres trouvé comme solution d'un problème d'estimation robuste est correcte ou non.

L'estimation est dite "robuste" dans le sens où ou le résultat ne doit pas être influencé par les données considérées comme erronées (i.e. abérantes), contrairement à d'autres méthodes, dont la plus connue

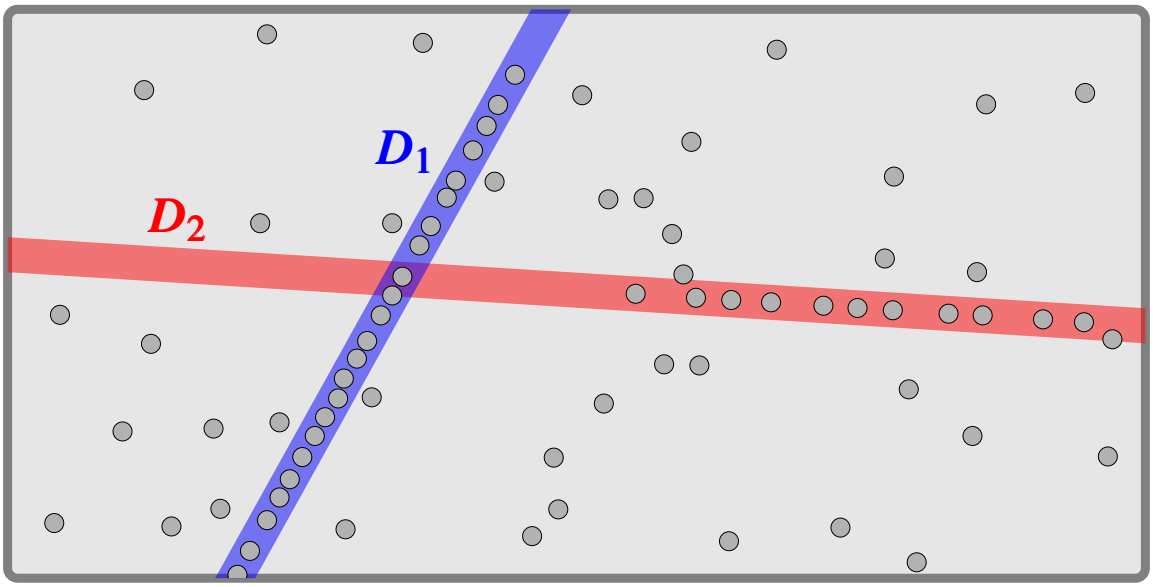


FIGURE 2.2 – Sortie dans le cas de l'exemple : les paramètres pour le modèle de droite : $\boldsymbol{D_1} = (a_1, b_1)$ et $\boldsymbol{D_2} = (a_2, b_2)$.

est les moindres carrés [62]. C'est pour cette raison que ce champ des mathématiques appliquées est particulièrement étudié dans le cadre d'applications en vision par ordinateur, où le taux d'erreurs est très variable et s'approche même parfois des 100%.

Nous détaillons dans la partie 7.1 comment notre problème de détection d'objets dans les images peut être ramené au problème d'estimation robuste.

2.1.3 Approches existantes

Il existe trois paradigmes bien distincts pour l'estimation robuste des paramètres d'un modèle : la transformé de Hough généralisée [1], RUDR [43] et RANSAC [19] Contrairement aux méthode par régression du type moindres carrés [62], ces méthodes estiment les paramètres du modèle recherché à partir d'un sous-ensemble des données, et non toutes les données. C'est cette propriété qui font leur intérêt dans le domaine de la vision par ordinateur, dont les capteurs sont souvent source de forts taux de données erronées.

2.1.3.1 Transformée généralisée de Hough

La transformée de Hough [1] est une façon de résoudre le problème de l'estimation des paramètres d'un modèle en restant entièrement dans l'espace des paramètres. Dans, le cas de l'estimation d'un mouvement entre deux images, chaque correspondance augmente le score de tous les ensembles de paramètres qui sont compatibles avec cette unique correspondance. Pour que cet ensemble ne soit pas infini, l'espace des paramètres est borné et discrétisé plus ou moins grossièrement. L'ensemble des paramètres obtenant le meilleur score est considéré comme le mouvement recherché. Du fait de la croissance exponentielle de la taille de l'espace discret des paramètres avec la complexité du modèle, cette méthode ne permet pas d'estimer précisément une homographie ou une géométrie épipolaire. Par contre, elle peut permettre un très bon "pré-nétoyage" des correspondances en identifiant les erreurs évidentes grâce à une discrétisation grossière de l'espace des paramètres.

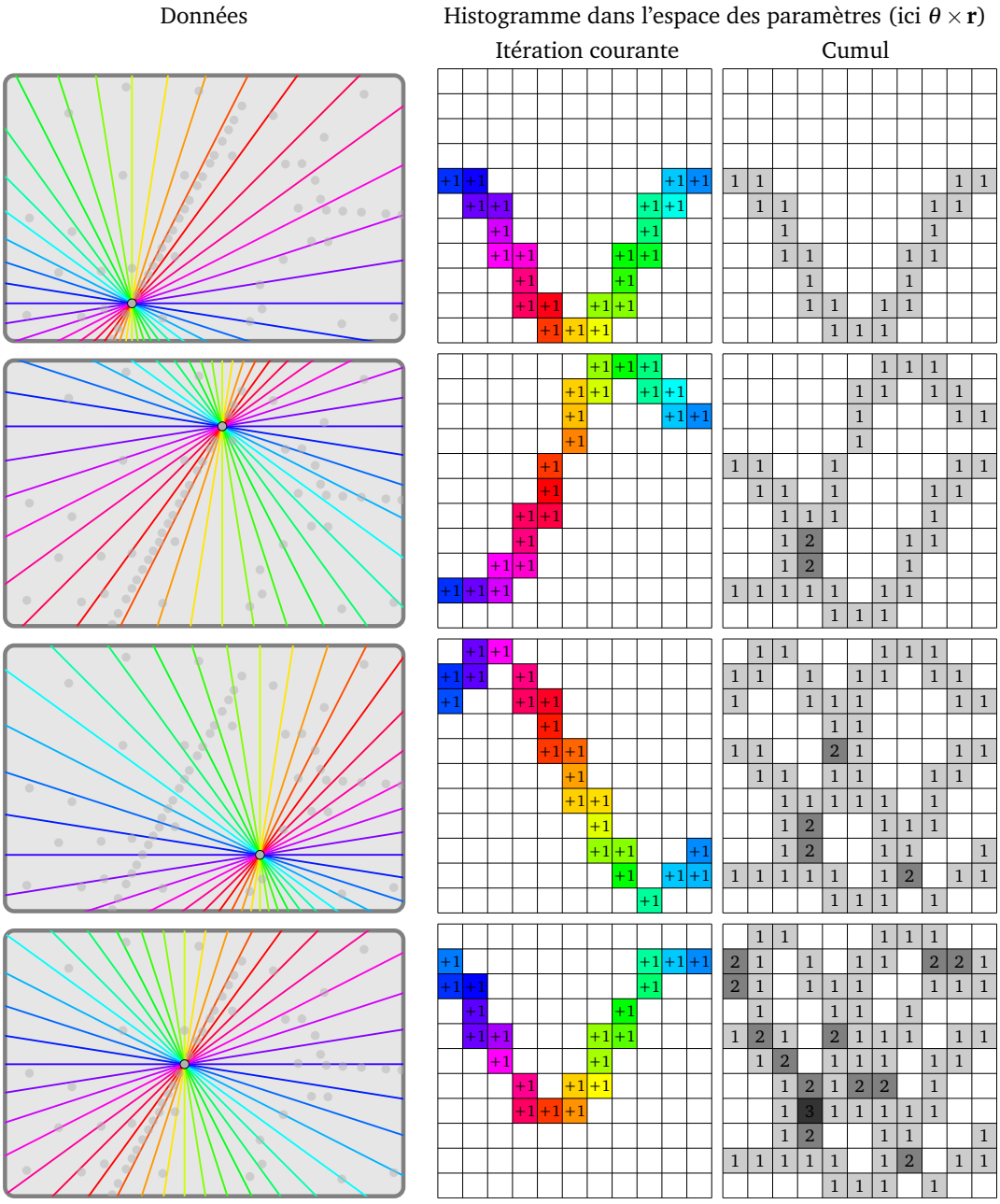


FIGURE 2.3 – Transformée de Hough : chaque donnée vote pour l'ensemble des cases de l'histogramme compatibles.

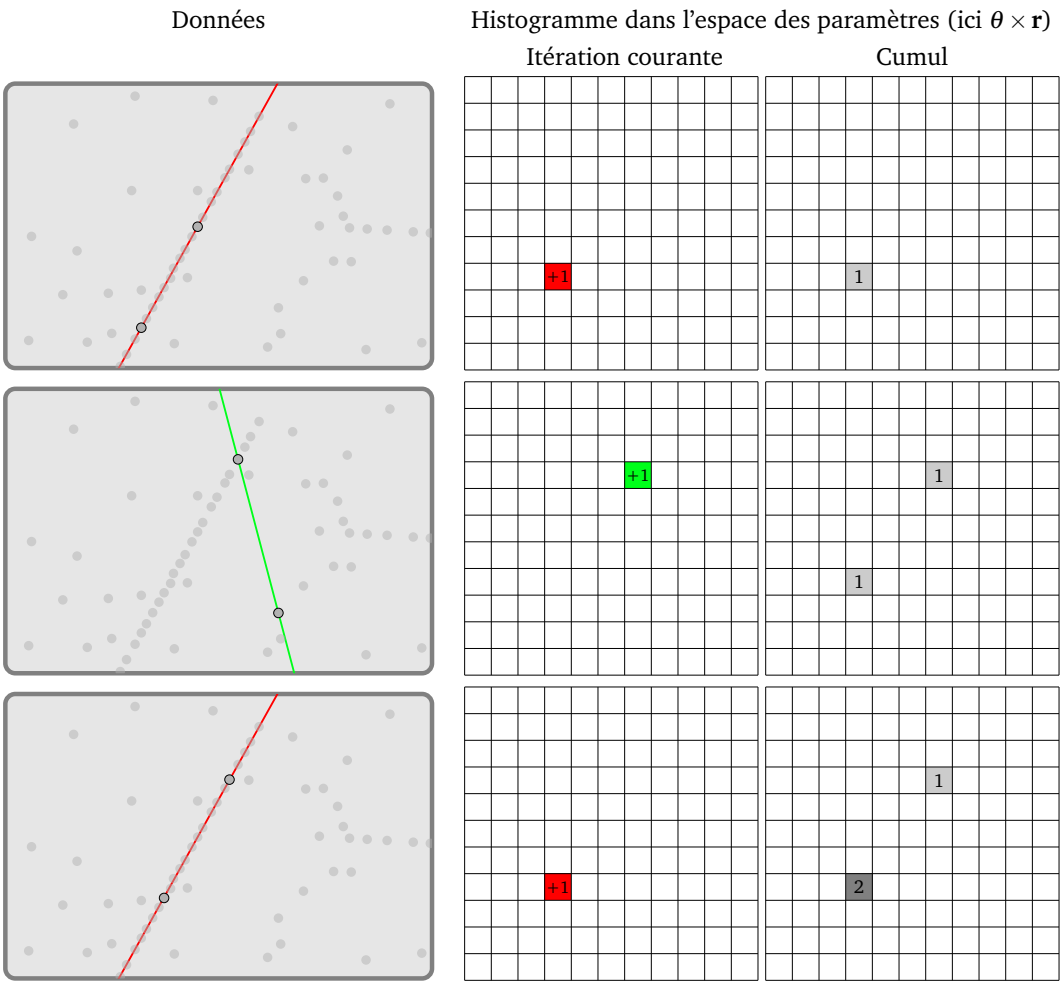


FIGURE 2.4 – RUDR : chaque ensemble suffisant de données (ici de taille 2) vote pour la case correspondant dans l'histogramme.

2.1.3.2 RUDR

RUDR (Recognition Using Decomposition and Randomization) [43], est de loin le paradigme le moins utilisé en vision par ordinateur. Cependant, Edwin Olson et al. [17] s'appuient dessus et sur les propriétés spectrales de la matrice d'adjacence d'un graphe pour former un groupe unique très cohérent dans l'espace des transformations. Les correspondances appartenant à ce groupe sont considérées comme correctes. L'approche de RUDR est de générer des hypothèses en utilisant un nombre de correspondances suffisant pour estimer le modèle, puis de les valider en les comparant directement entre elles dans l'espace des paramètres du modèle (comme dans la transformée de Hough).

2.1.3.3 RANSAC

Les nombreuses méthodes dites *Generate and test* sont celles du groupe RANSAC (RANdom SAmple Consensus) [19]. L'idée est d'estimer les paramètres du modèle grâce à un sous-ensemble des correspondances et de vérifier le modèle sur l'ensemble des données. L'algorithme original, publié par M. Fischler & R. Bolles en 1981 est présenté figure 2.5. Dans le chapitre suivant, nous détaillons précisément chaque étape de RANSAC.

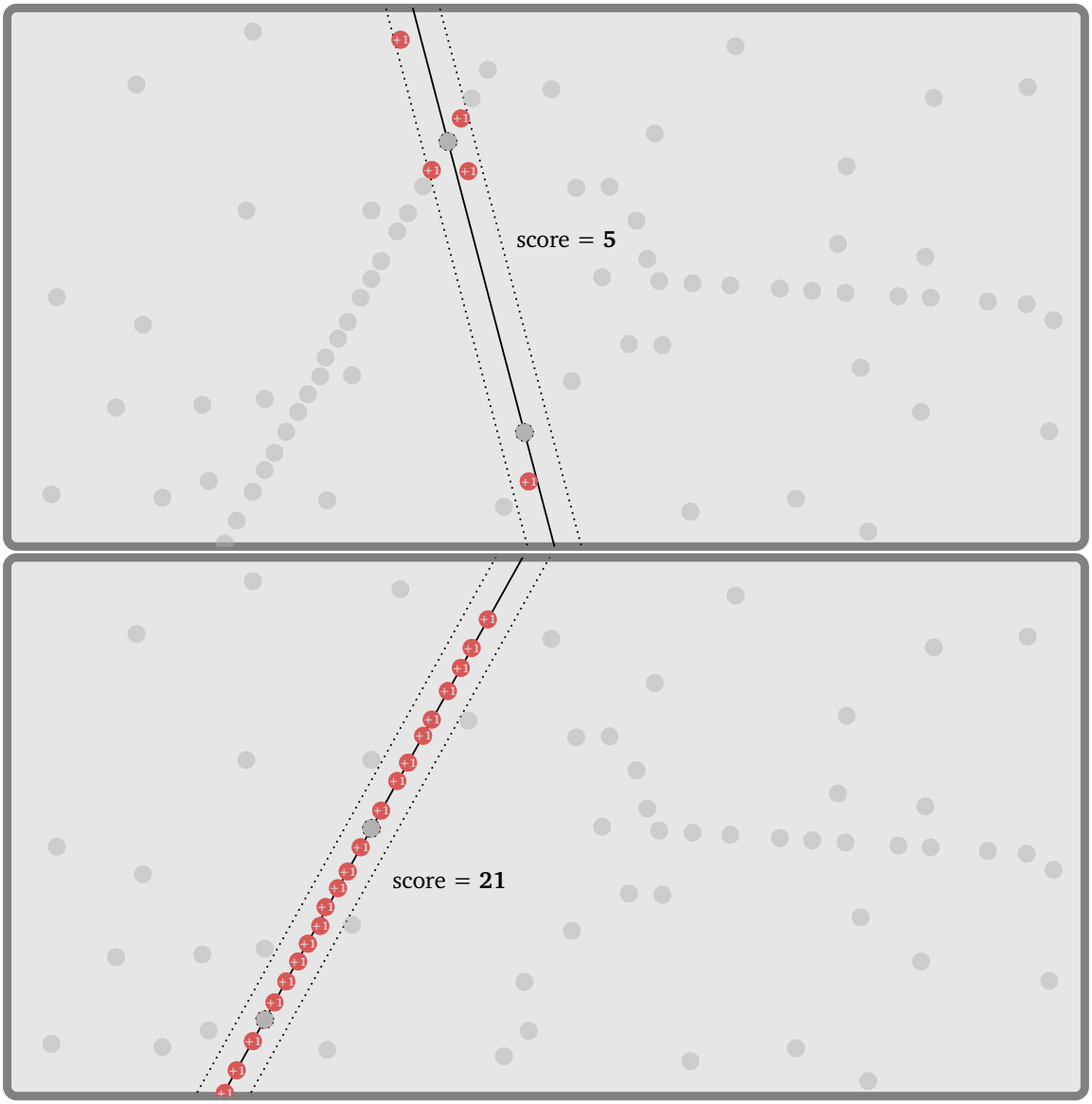


FIGURE 2.5 – RANSAC : tout se passe dans l'espace des données. A chaque itération, un ensemble suffisant de données est sélectionné au hasard (ici de taille 2), son score est évalué en comptant les autres données compatibles.

2.2 Analyse détaillée de RANSAC

RANSAC (RANdom Sample Consensus) [19] est un algorithme non déterministe pour l'estimation des paramètres d'un modèle mathématique à partir de N données en partie erronées. Une donnée erronée est une donnée dont la présence n'est pas explicable par le modèle recherché, mais par un autre modèle, le plus souvent inconnu. Les autres données, celles qui sont générées par le modèle que l'on souhaite estimer, contiennent une part plus ou moins importante d'erreur, appelée erreur de mesure ou bruit.

L'algorithme est composé de trois étapes répétées itérativement jusqu'à ce qu'une condition d'arrêt devienne vraie :

1. **Création d'une hypothèse.** Un échantillon de taille m parmi N données est sélectionné aléatoirement. Les paramètres du modèle sont calculés à partir de cet échantillon. m est le nombre minimum de données nécessaires pour ajuster le modèle.
2. **Évaluation de l'hypothèse.** On compte le nombre de données compatibles avec le modèle précédemment estimé.
3. **Test de la condition d'arrêt.** Un calcul est effectué. Si la probabilité de trouver une meilleure solution (un modèle en accord avec plus de données) tombe sous un seuil prédéfini, les itérations s'interrompent.

Nous présentons cet algorithme de manière graphique dans le schéma 2.2. Chaque étape est détaillée dans les paragraphes qui suivent.

2.2.1 Création d'une hypothèse

La création d'une hypothèse pour l'ensemble des paramètres du modèle recherché est très simple. RANSAC sélectionne un nombre m de données parmi les N données du problème selon un hasard uniforme. Chaque donnée a ainsi une chance $\frac{m}{N}$ d'être sélectionnée pour construire l'hypothèse en cours. Le nombre m de données sélectionnées est le plus petit nombre nécessaire à l'estimation des paramètres, c'est pourquoi on peut appeler m la complexité du modèle recherché.

Si l'on prend l'exemple de l'estimation d'une droite dans un nuage de points à deux dimensions, alors nos données sont l'ensemble des points et notre modèle peut par exemple être l'équation de droite

$$\mathbf{a} \cdot x + \mathbf{b} \cdot y + 1 = 0,$$

où **a** et **b** sont les paramètres à estimer. Ce modèle a une complexité de $m = 2$ car il faut au moins deux points pour estimer les paramètres d'une droite. Pour estimer ces deux paramètres, il faut donc choisir un échantillon de deux données, appelé échantillon-hypothèse. À partir d'un échantillon hypothèse $\{(x_{h_1}, y_{h_1})$, $(x_{h_2}, y_{h_2})\}$ ont peut déduire une paramétrisation-hypothèse (a_h, b_h) pour notre modèle en résolvant le système 2.1. Ces étapes sont schématisées dans la figure 2.7.

$$\begin{cases} \mathbf{a_h} \cdot x_1 + \mathbf{b_h} \cdot y_1 + 1 = 0 \\ \mathbf{a_h} \cdot x_2 + \mathbf{b_h} \cdot y_2 + 1 = 0 \end{cases} \tag{2.1}$$

Pour simplifier, dans la suite nous parlerons souvent (abusivement) de "modèle-hypothèse" pour "paramétrisation-hypothèse".

2.2.2 Évaluation d'une hypothèse

Pour évaluer un modèle-hypothèse, le nombre de données compatibles avec celui-ci est compté, en les testant une à une. Une donnée est considérée comme compatible avec le modèle si sont "écart" (ou, erreur résiduelle) avec celui-ci est en dessous d'une limite prédéfinie. L'écart entre une donnée et un modèle doit être défini selon l'application.

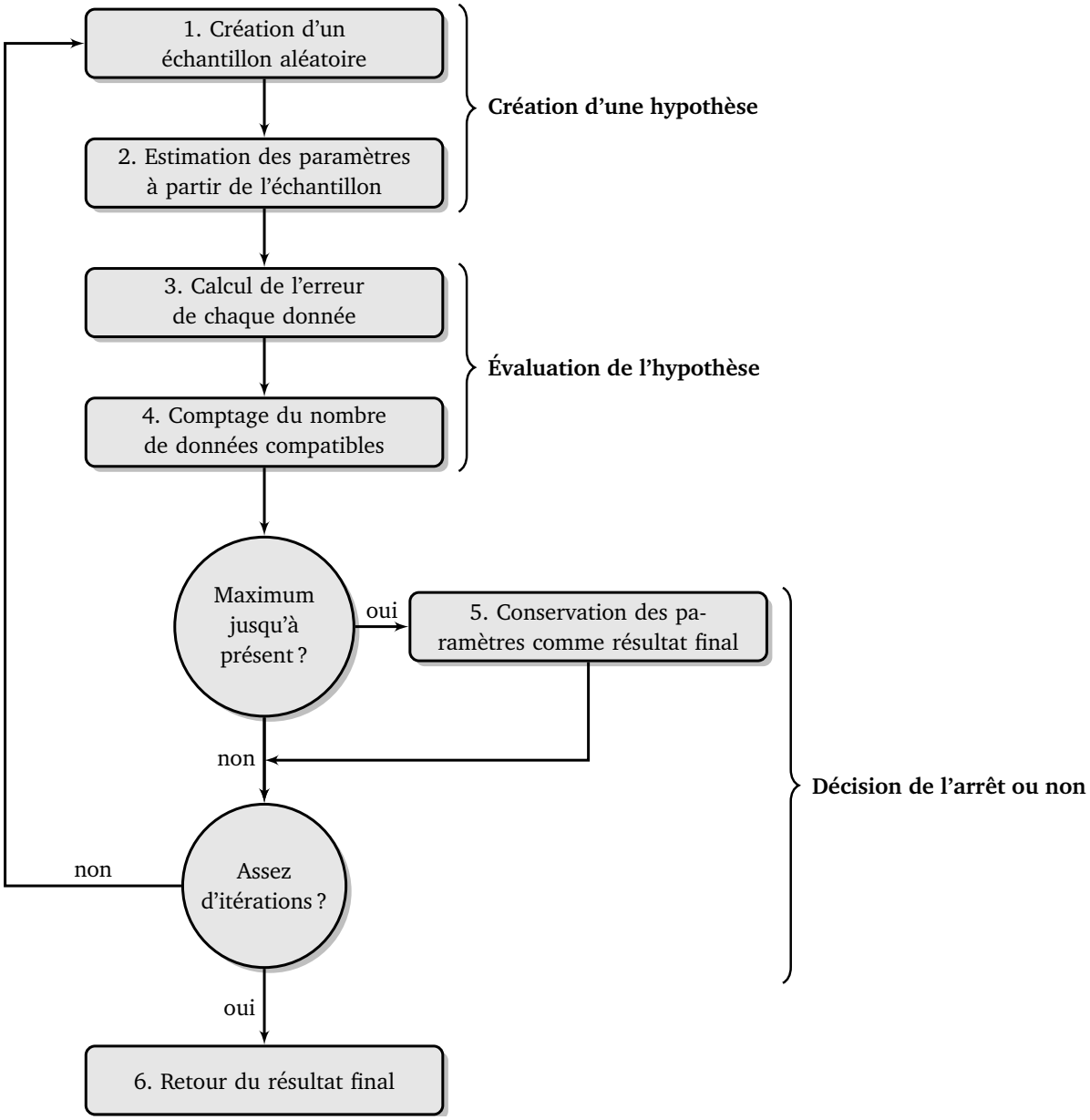


FIGURE 2.6 – Schéma de l'algorithme RANSAC

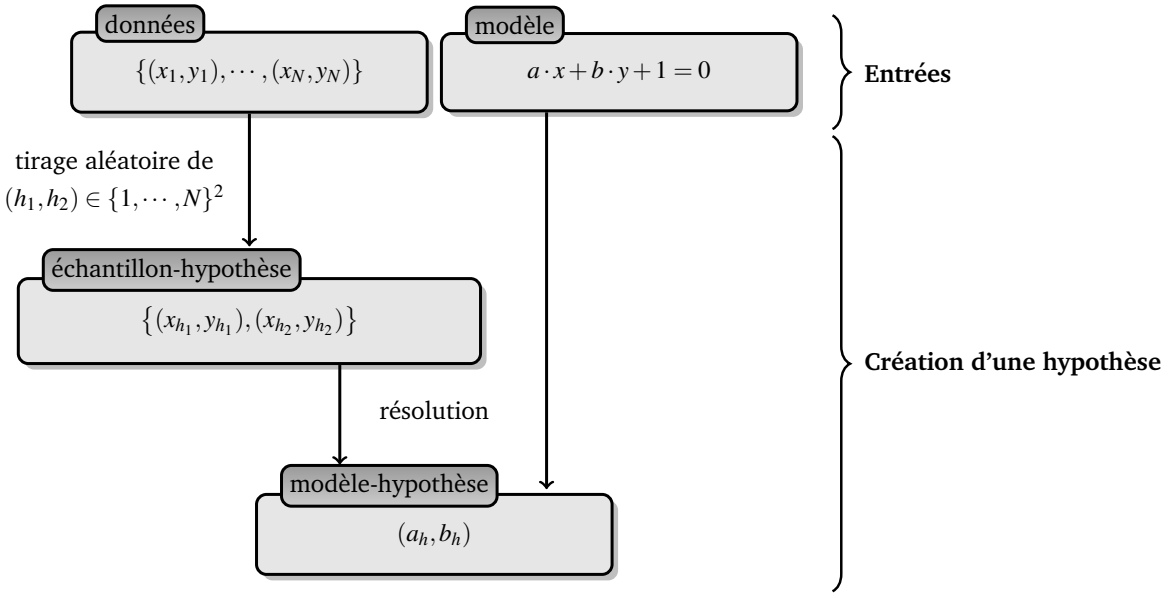


FIGURE 2.7 – Création d'un modèle-hypothèse dans le cas de l'estimation d'une droite dans un nuage de points en deux dimensions.

Dans le cas ne notre modèle de droite en deux dimensions, on peut par exemple définir l'écart entre une droite-hypothèse $a\cdot x+b\cdot y+1=0$ et un point (x,y) comme la distance euclidienne séparant le point de son projeté orthogonal sur la droite (équation 2.2).

$$écart((x,y),(a,b)) = \frac{|a\cdot x+b\cdot y+1|}{\sqrt{a^2+b^2}} \tag{2.2}$$

2.2.3 Condition d'arrêt

RANSAC s'arrête lorsque que le nombre d'itérations effectuées devient suffisant pour que la probabilité de trouver une meilleure hypothèse (que la meilleure trouvée jusqu'à présent) tombe sous un certain seuil μ. Soient N, m et I, respectivement le nombre total de données, le nombre de données nécessaires pour instancier un modèle-hypothèse et le taux de données correctes.

La probabilité P_I de sélectionner un échantillon de taille m non contaminé (i.e. ne contenant pas de donnée erronée) vaut :

$$P_I = \frac{\begin{pmatrix} I\cdot N \\ m \end{pmatrix}}{\begin{pmatrix} N \\ m \end{pmatrix}} = \prod_{j=0}^{m-1} \frac{I\cdot N-j}{N-j} \approx I^m \tag{2.3}$$

On en déduit μ, la probabilité de ne former aucun échantillon non contaminé après k itérations,

$$\mu = (1-P_I)^k \tag{2.4}$$

et le nombre, k_{μ_0}, d'échantillons à former pour tomber sous un seuil μ_0 prédéfini.

$$k_{\mu_0} \geq \frac{ln(\mu_0)}{ln(1-P_I)} \tag{2.5}$$

On choisit généralement $\mu_0 = 1\%$. La décision de l'arrêt ou non des itérations nécessite donc de connaître le taux de données correctes I. En pratique, on ne connaît jamais exactement la valeur de I mais on

sait que $I \geq I_{observé}$, où $I_{observé}$ est le taux de données compatibles avec le meilleur modèle trouvé jusqu'à présent.
Ainsi,

$$\frac{ln(\mu_0)}{ln\left(1-P_{I_{observé}}\right)} \geq \frac{ln(\mu_0)}{ln(1-P_I)}$$

donc on impose :

$$k_{\mu_0} \geq \frac{ln(\mu_0)}{ln\left(1-P_{I_{observé}}\right)} \tag{2.6}$$

RANSAC s'arrête lorsque le nombre d'itérations dépasse la valeur k_{μ_0}. Ce critère ne garantit évidemment pas d'obtenir le meilleur modèle, pour cela il faudrait faire une recherche exhaustive, de complexité $O\left(N^{(m+1)}\right)$.

2.3 Approches existantes pour améliorer RANSAC

Depuis trente ans, de très nombreuses méthodes ont été dérivées de RANSAC en vision par ordinateur. Chaque étape de l'algorithme s'est vu proposé des améliorations. Dans ce chapitre, nous présentons celles qui ont montré les meilleurs résultats.

2.3.1 Améliorations de l'échantillonnage

Une partie des approches pour améliorer RANSAC se concentre sur l'échantillonnage (i.e. la sélection des données pour former une hypothèse). Dans RANSAC, il est purement uniforme. Dans cette section, nous distinguons deux types d'échantillonnage non uniforme que nous appelons l'*échantillonnage biaisé* et l'*échantillonnage réordonné*.

2.3.1.1 Échantillonnage biaisé

Une approche possible est de biaiser l'échantillonnage pour augmenter la probabilité de former des échantillons sans données erronées. Le biais est souvent basé sur des informations additionnelles donnant un *a priori* sur la validité d'une donnée observée.

Guided-MLESAC Guided-MLESAC [55] utilise une probabilité de validité associée à chaque donnée. La chance qu'une donnée a d'être tirée pour former un échantillon-hypothèse est proportionnelle à sa probabilité d'être correcte. De plus, MLESAC est capable de gérer un (unique) objet en mouvement en estimant la probabilité d'un point de faire partie du fond ou de l'objet. Le problème de cette approche, comme beaucoup d'autres, est qu'elle nécessite des valeurs de probabilités alors qu'en pratique, on dispose d'une information assez faibles dont il est difficile de dériver des probabilités.

BaySAC et SimSAC BaySAC et SimSAC [4] utilisent le résultat des itérations précédentes comme source additionnelle d'information. Les données composant les échantillons ayant échoué précédemment ont moins de chance d'être tirées pour former de nouvelles hypothèses. Les probabilités déduites des itérations précédentes sont cependant assujetties à de lourdes hypothèses qui rendent le calcul soluble. L'inconvénient majeur de ces techniques est que l'échantillonnage devient déterministe au cours des itérations, sans qu'elles n'apportent aucune garantie sur l'impossibilité de "tourner en boucle" en sélectionnant constamment les mêmes données. D'ailleurs, les auteurs reconnaissent que cet échantillonnage peut échouer systématiquement pour certaines "configurations dégénérées".

Guided sampling via weak motion models Dans le cas de l'estimation d'une matrice fondamentale, l'instanciation d'un modèle-hypothèse nécessite la sélection de 7 ou 8 correspondances selon la méthode. Or, il suffit qu'une seule de ces correspondances soit erronée pour que l'hypothèse soit très éloignée des vrais paramètres. L'idée de la méthode présentée dans [21], est alors d'approcher le mouvement avec un modèle beaucoup moins complexe (celui de la transformation affine), nécessitant seulement trois correspondances. En générant des hypothèses plus faibles, la méthode déduit une probabilité pour une correspondance d'être correcte et utilise ces probabilités pour guider RANSAC. Il est aussi possible de déduire le taux de données correctes, ce qui est utile pour affiner divers paramètres de RANSAC en ligne.

NAPSAC (N Adjacent Points SAmple Consensus) NAPSAC [40] utilise le fait que si deux correspondances partent de deux points proches dans la première image et arrivent sur deux points proches, eux aussi, dans la seconde image, alors ce sont probablement deux correspondances correctes. La méthode est alors très simple : la première correspondance de l'échantillon est choisie uniformément et les suivantes sont choisies parmi les plus proches dans l'espace à quatre dimensions correspondant à la concaténation des coordonnées des deux points des correspondances. Ce type d'approches, basées sur la proximité dans l'espace image, possèdent toute le défaut d'être particulièrement sensibles à l'imprécision de la position des points d'intérêt. De plus, elles ne supportent pas un grand changement d'échelle entre les deux images.

J-Linkage A l'instar de NAPSAC, J-Linkage [53] propose un processus d'échantillonnage favorisant la sélection de points voisins. En effet, si un point x_i a déjà été sélectionné, alors la probabilité de sélectionner x_j est

$$P(x_j \mid x_i) = \begin{cases} \frac{1}{Z}\exp(-\frac{\|x_j - x_i\|^2}{\sigma^2}) & \text{si } x_j \neq x_i \\ 0 & \text{si } x_j = x_i \end{cases},$$

où σ doit être choisi expérimentalement, selon les données. J-Linkage tire M échantillons de cette façon, puis en calcule un partitionnement dont la distance entre les échantillons est la distance de Jaccard (équation 2.7) entre les ensembles de données en accord avec les échantillons.

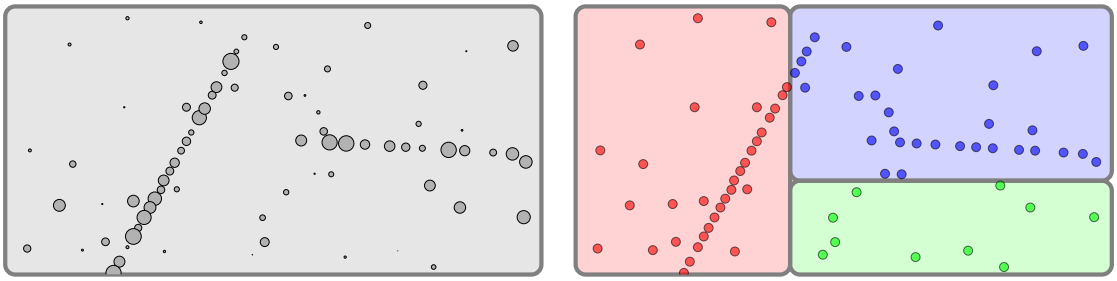
$$d(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{2.7}$$

Cet échantillonnage comporte de nombreux inconvénients : la variable aléatoire de sélection nécessite la connaissance des distances entre toutes les données (ce qui représente un long calcul en $O(N^2)$), et le nombre d'échantillons M à effectuer est nécessairement élevé. De plus, les paramètres σ et M sont très dépendant des données et doivent pourtant être définis par l'utilisateur.

RANSAC 3-LAF & RANSAC 2-LAF L'idée de ces méthodes [13] [45] est de créer plusieurs correspondances à partir d'une correspondance entre deux patches similaires à une transformation affine près. Les correspondances ainsi crées sont très proches par rapport à leur position spatiale, ce qui nécessite une étape indispensable d'optimisation issue de LO-RANSAC [12]. Mais, le gain de temps est considérable car la géométrie épipolaire peut ainsi être estimée à partir de petits échantillons de 2 ou 3 correspondances, qui ont évidemment beaucoup moins de chances d'être contaminés qu'un échantillon de 7 ou 8 correspondances.

2.3.1.2 Échantillonnage réordonné

Si biaiser l'échantillonnage peut accélérer RANSAC, cela peut également affecter grandement la recherche si l'information utilisée n'est pas fiable, ou encore si la randomisation devient insuffisante. C'est pourquoi des méthodes proposent de limiter le guidage à un réordonnancement des échantillons formés.



Perception de PROSAC Perception de GroupSAC

FIGURE 2.8 – Exemple de représentation de la perception des méthodes PROSAC et GroupSAC

Ainsi, malgré un minimum d'hypothèses sur l'information utilisée, PROSAC [11] et GroupSAC [42] s'assurent de ne jamais être pires qu'une sélection purement aléatoire. Lors des T premières itérations, les échantillons sont formés en commençant par les plus probables et en finissant par les moins probables. Après cette phase, chaque échantillon a eu la même chance d'être formé, mais les échantillons considérés comme meilleurs ont plus de chances d'être formés tôt. Ensuite, l'échantillonnage est uniforme. En choisissant T comme le nombre moyen d'itérations nécessaires à RANSAC pour trouver un bon modèle, ces deux méthodes font, dans le pire cas, comme RANSAC. La figure 2.8 illustre la perception de ces deux méthodes, que nous détaillons succinctement plus bas et plus en détail dans le paragraphe 3.2.

PROSAC PROSAC [11] commence par ordonner toutes les données, des plus sûres au moins sûres, d'après une certaine confiance *a priori*. L'*a priori* utilisé dans l'article est le score de mise en correspondance des points détectés par SIFT [31]. Ce score est égal à la distance entre les vecteurs SIFT des deux points mis en correspondance, divisée par la distance entre le premier point et son second plus proche vecteur SIFT dans la seconde image. Un rapport faible signifie que les vecteurs SIFT des deux point sont très proches l'un de l'autre par rapport aux autres. Donc, plus il est faible, plus la correspondance a de chances d'être correcte. Une fois les données ainsi ordonnées, l'algorithme est très simple : il s'agit d'appliquer RANSAC au groupe des données les mieux classées. La taille de ce groupe augmente progressivement au cours des itérations, jusqu'à ce qu'une solution au problème soit trouvée.

GroupSAC GroupSAC [42], quant à lui, n'utilise pas une confiance associée à chaque donnée. Il considère les données par groupes, et, comme démontré par les auteurs, sous très peu d'hypothèses les échantillons formés de données issues de peu de groupes différents ont moins de chances d'être contaminés que les autres. Alors le nombre de groupes représentés dans chaque échantillon formé est petit dans les premières itérations et s'agrandit tant qu'aucune solution n'est trouvée. Dans le cas de la vision par ordinateur, les auteurs proposent de former les groupes en fonction du flux optique (dans le cas d'une vidéo) ou d'une segmentation des images basée sur la couleur et la texture. Ce type de groupement peut être très informatif mais aussi souvent très coûteux.

2.3.2 Améliorations de la vérification

La vérification d'une hypothèse sur l'ensemble des données est l'étape la plus coûteuse de RANSAC en temps. Alors de nombreuse méthodes proposent d'accélérer cette étape, quitte à augmenter le nombre d'itérations en contrepartie.

2.3.2.1 Évaluation partielle

Pour gagner du temps, une stratégie est d'interrompre l'évaluation d'une hypothèse lorsqu'elle paraît peut probable.

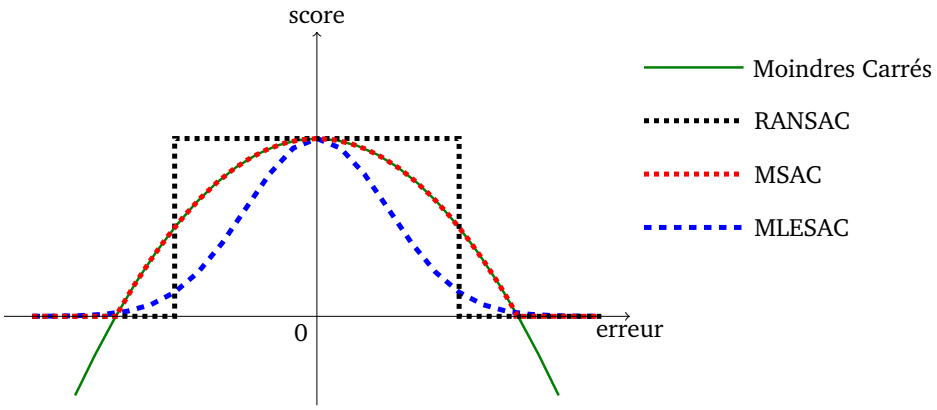


FIGURE 2.9 – Fonctions de perte utilisées par quatre méthodes différentes. On voit clairement ce qui différencie les moindres carrés des méthodes dites robustes : sa fonction de perte n'est pas bornée.

R-RANSAC (Randomized RANSAC) R-RANSAC [9] (Randomized RANSAC) effectue un test préliminaire, le $T_{d,d}$ test. Une évaluation complète de l'hypothèse est effectuée seulement si elle passe ce test. Si ce n'est pas le cas, l'hypothèse est rejetée (potentiellement à tort). Le $T_{d,d}$ test est passé si d données sélectionnées au hasard sont toutes cohérentes avec l'hypothèse. Les auteurs montrent que $d = 1$ est la valeur optimale.

Bail-out test De façon très similaire, un RANSAC avec un test *Bail-out* a été proposé [33]. L'évaluation d'une hypothèse s'effectue sur un échantillon des données et s'interrompt lorsque assez de données ont été utilisées pour être suffisamment confiant dans l'inutilité de l'hypothèse. Dans ce cas elle est rejetée.

SPRT (Sequential Probability Ratio Test) Un autre R-RANSAC [34] utilise le *Sequential Probability Ratio Test* (SPRT). Il est une version optimale du *Bail-out test*. Il s'appuie sur la théorie de la prise de décisions séquentielles de Wald, qui permet d'optimiser le choix de continuer ou d'arrêter la vérification d'une hypothèse.

2.3.3 Améliorations de la précision

Des méthodes cherchent à améliorer la précision du résultat. Pour ce faire, certaines d'entre elles modifient la *fonction de perte*, c'est-à-dire le gain de score apporté à l'hypothèse, par chaque donnée en fonction de son erreur résiduelle. La figure 2.9 présente quatre fonctions de perte différentes. La fonction, dépendant de l'erreur e, utilisée par RANSAC est une simple fonction binaire :

$$Perte_{RANSAC}(e) = \begin{cases} 1 & \text{si } |e| < seuil \\ 0 & \text{sinon} \end{cases}$$

2.3.3.1 MSAC (M-estimator SAmple Consensus)

MSAC [56] utilise une fonction de perte quadratique bornée :

$$Perte_{MSAC}(e) = \begin{cases} e^2 & \text{si } |e| < c \\ c^2 & \text{sinon} \end{cases}$$

2.3.3.2 MLESAC (Maximum Likelihood SAmple Consensus)

MLESAC [56] utilise deux fonctions de densité de probabilité pour modéliser l'erreur des bonnes et des mauvaises données. L'erreur des bonnes données est modélisée pour une gaussienne centrée, et l'erreur des données erronées est modélisée par une variable uniforme.

$$P(e|M) = \gamma \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e^2}{2\sigma^2}\right) + (1-\gamma)\frac{1}{v}$$

L'hypothèse choisie par MLESAC est celle qui possède la vraisemblance la plus grande selon ce modèle.

$$Perte_{MLESAC}(e) = -ln(p(e|M))$$

2.3.3.3 LO-RANSAC (Locally Optimized RANSAC)

Une fonction de perte complexe représente un coût supplémentaire en temps de calcul non négligeable. LO-RANSAC [12] propose d'affiner une hypothèse seulement si elle obtient un score élevé avec la fonction de perte de RANSAC. Si c'est le cas, alors plusieurs stratégies plus où moins coûteuses sont proposées pour calculer un score plus fin et faire converger la solution de façon à maximiser ce score. Seules les données en accord avec l'hypothèse sont prises en compte pour conserver la robustesse de RANSAC.

2.3.4 Amélioration de la robustesse

Selon le problème et les données sur lesquelles on travaille, les données correctes et erronées peuvent être distribuées de façons très différentes. Il est donc nécessaire, pour obtenir une robustesse sans connaissances fortes sur les propriétés des données, d'être capable de s'adapter.

2.3.4.1 Évaluation adaptative

MAPSAC & u-MLESAC MAPSAC (Maximum A Posteriori SAmpleConsensus) [18] et u-MLESAC [10] toutes deux dérivées de MLESAC, en estiment les espérances σ et γ de façon itérative par la méthode EM (Expectation Maximisation).

AMLESAC [57] est très similaire aux deux méthodes précédentes. Elle se différencie par le sa recherche de σ, qui est réalisée par une descente de gradient.

2.3.4.2 Terminaison adaptative

MAPSAC Dans MAPSAC, le nombre d'itérations à effectuer est calculé comme dans RANSAC (équation 2.5), mais il doit être mis à jour au cours des itérations avec la valeur de γ estimée.

u-MLESAC u-MLESAC propose également d'adapter le nombre d'itérations à effectuer, t, à partir des valeurs de γ et σ estimées (équation 2.8, où erf désigne la fonction d'erreur de Gauss).

$$t = \frac{ln(\alpha)}{ln(1-k^m\gamma^m)} \quad \text{avec} \quad k = erf(\frac{\beta}{\sqrt{2}\sigma}) \tag{2.8}$$

2.3.5 Sélection du modèle

Certaines méthodes, parmi les moins supervisées, ne se contentent pas de déterminer les paramètres d'un modèle, elle sont également capables de sélectionner le meilleur modèle parmi plusieurs connus.

2.3.5.1 Méthodes *a contrario*

Les méthodes dites *a contrario* [7, 37, 47] fournissent une solution parfaitement adaptative au problème de l'estimation robuste d'un modèle grâce à une approche sans paramètre. L'idée est simple : se donner un modèle du hasard et rechercher des anomalies par rapport au hasard dans les données. Ces anomalies, non explicables par le modèle du hasard, sont générées par les modèles recherchés. Cette approche est appliquée dans le pipeline complet, de la génération des données à la sélection du bon modèle et de ses parametres.

2.3.5.2 QDEGSAC

QDEGSAC (RANSAC for Quasi-Denerated Data) [20] introduit un RANSAC supplémentaire au cœur de l'algorithme, qui cherche un modèle de moindre dimension pour détecter une configuration dégénérescente. Typiquement, un plan contenant beaucoup de points d'intérêt sera détecté et sous-utilisé pour l'estimation d'une géométrie épipolaire.

2.4 Motivations et étude préliminaire

2.4.1 Défauts de RANSAC

2.4.1.1 Nombre d'itérations

Le nombre d'itérations théorique nécessaires à RANSAC pour trouver une solution s'accroît très vite avec le taux de données erronées $(1-I)$ et la complexité du modèle, m. Pour qu'un modèle proche de celui que l'on cherche soit estimé, il faut tirer un échantillon-hypothèse dont toutes les données sont correctes, i.e. expliquées/générées par ce modèle. Nous verrons dans le paragraphe suivant que cette condition nécessaire n'est cependant pas suffisante. Le tableau 2.1 est une application numérique du nombre d'itérations que doit réaliser RANSAC pour résoudre différents problèmes (équation 2.6), en acceptant une probabilité d'échouer μ_0 égale à 1%.

$k_{1\%}(\boldsymbol{m},\boldsymbol{I})$

$\boldsymbol{m}$ \ $\boldsymbol{I}$	80%	60%	40%	20%	15%	10%	5%
4	9	33	178	$2.9\cdot10^3$	$9.1\cdot10^3$	$4.6\cdot10^4$	$7.4\cdot10^5$
7	20	162	$2.8\cdot10^3$	$3.6\cdot10^5$	$2.7\cdot10^6$	$4.6\cdot10^7$	$5.9\cdot10^9$

TABLE 2.1 – Nombre d'échantillons de m données qu'il faut former pour avoir la probabilité $1-\mu_0 = 99\%$ d'en avoir formé un ne contenant pas d'erreur, en fonction du taux de données correctes I.

2.4.1.2 Correction et pertinence

La correction d'une donnée ou d'un échantillon-hypothèse n'implique pas sa pertinence, loin de là. Le nombre d'itérations effectuées par RANSAC repose pourtant sur cette hypothèse. Ce fait est très souvent ignoré dans la littérature alors qu'il est très loin d'être négligeable.

Une donnée Les données considérées comme correctes, sont générées par le modèle dont on recherche les paramètres, mais pas uniquement. Un second modèle affecte leurs valeurs, c'est l'erreur de mesure. L'erreur de mesure est une composante aléatoire additionnée aux valeurs issues du modèle, qui est toujours présente mais dont l'explication et l'ampleur dépend bien sûr de l'application. Ainsi, en pratique,

il n'existe jamais de segmentation parfaite en deux groupes des données : les données correctes et les données erronées. Au lieu de cela, toutes les données sont plus ou moins bonnes.

La figure 2.11 montre à quel point une application pratique peut être éloignée de la situation idéale théorique supposée par RANSAC.

Un échantillon de données Si une donnée peut être plus ou moins bonne, c'est aussi le cas d'un échantillon, d'une façon encore plus importante. En effet, dans beaucoup d'applications un échantillon complet, composé uniquement de données correctes n'est pas forcément suffisant pour estimer les paramètres du modèle. En particulier, l'échantillon ne doit pas être dégénéré, dans le sens où il est explicable par un modèle plus simple que celui que l'on recherche. Par exemple, 4 points d'intérêt alignés ou 8 points d'intérêt coplanaires ne permettent pas d'estimer respectivement une homographie ou une matrice fondamentale. Plus le bruit de mesure est important, plus les échantillons doivent être éloignés d'une configuration dégénérée pour fournir une estimation correcte du modèle recherché. D'une façon générale et triviale, mais qui va à l'encontre de nombreuses méthodes [41, 53, 42], plus les points d'intérêt d'un échantillon sont éloignés les uns des autres dans l'espace image, plus l'estimation aura de chances d'être précise.

Coexistance de plusieurs modèles Dans le cas où plusieurs instances du modèle recherché coexistent dans les données, alors, il est évident que tout échantillon formé de données appartenant à des instances différentes ne permettra de retrouver aucune d'entre elles. En supposant que les instances contiennent le même nombre de données, le tableau 2.2 montre à quel point la probabilité de former par hasard un échantillon utile s'effondre très rapidement avec le nombre d'instances.

Taille de l'échantillon \ Nombre d'instances	**1**	**2**	**3**	**4**
4	100%	12.5%	3.7%	1.56%
7	100%	1.56%	0.14%	0.02%

TABLE 2.2 – Probabilité de former par hasard un échantillon composé d'une seule instance dans le cas où plusieurs instances coexistent. On suppose qu'il n'y a pas de donnée erronée ($I = 100\%$) et que chaque instance est composée du même nombre de données.

2.4.2 Sources d'information sous exploitées

Pendant notre étude de l'état de l'art, un constat nous est rapidement apparu évident : RANSAC est toujours appliqué à une partie éparse et très incomplète de l'information du problème, typiquement un ensemble de correspondances de points d'intérêt. Le fait que les données soient éparses est normal, c'est le seul type d'information que RANSAC peut traiter. Pourtant, dans toutes ses applications courantes en vision par ordinateur, les données sont initialement denses, puisque issues d'images ou de vidéos, donc infiniment plus riche en information. Dans cette partie, nous faisons un tour de cette information, que nous considérons sous exploitée.

2.4.2.1 Informations issues du signal

Comme nous l'expliquons dans le paragraphe 7.1.1.2, l'information contenue dans le signal de l'image ou de la vidéo est largement utilisée pendant la phase de description des caractéristiques en vue de la mise en correspondance des caractéristiques les plus proches visuellement. Mais, dans la phase d'estimation

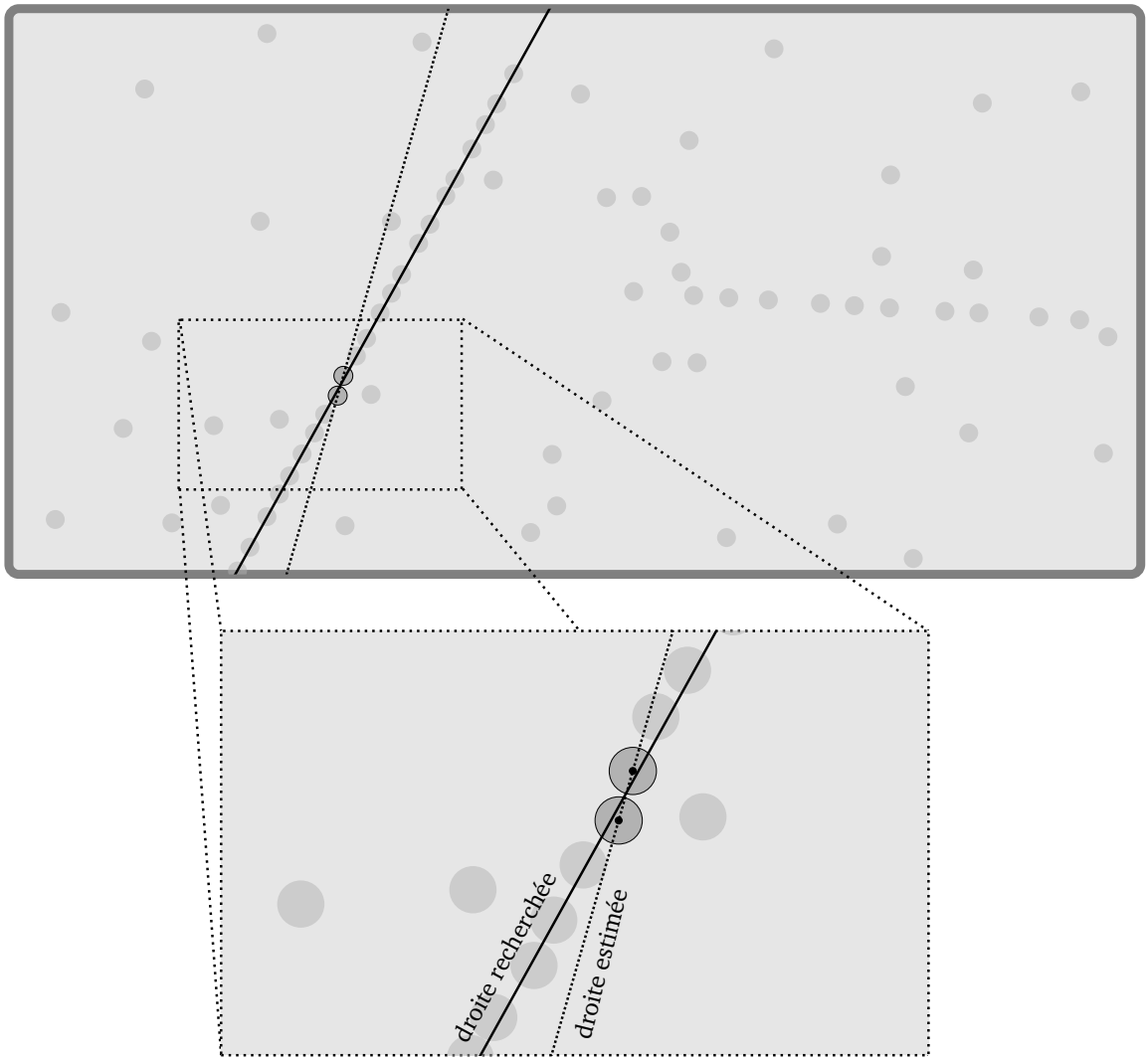


FIGURE 2.10 – La droite estimée à partir de données trop proches peut être très éloignée de la droite recherchée.

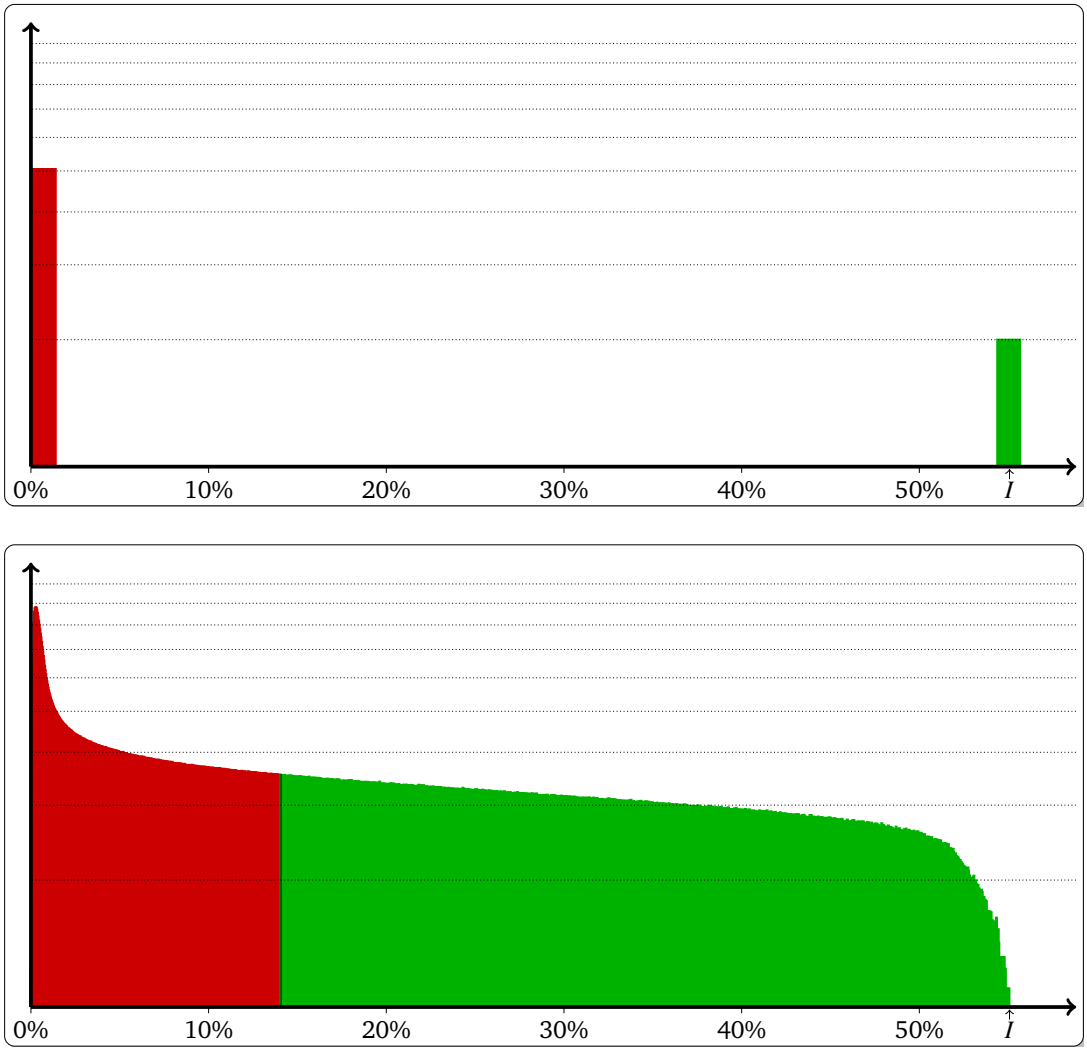


FIGURE 2.11 – Histogramme du score des échantillons formés selon un hasard uniforme, en échelle logarithmique. En haut : le cas parfait, l'hypothèse faite par RANSAC. En bas : un cas réel typique. Le taux d'échantillons non contaminés est I^m. Ce graphique met en exergue l'éloignement entre la réalité et le modèle parfait sous-jacent à RANSAC.

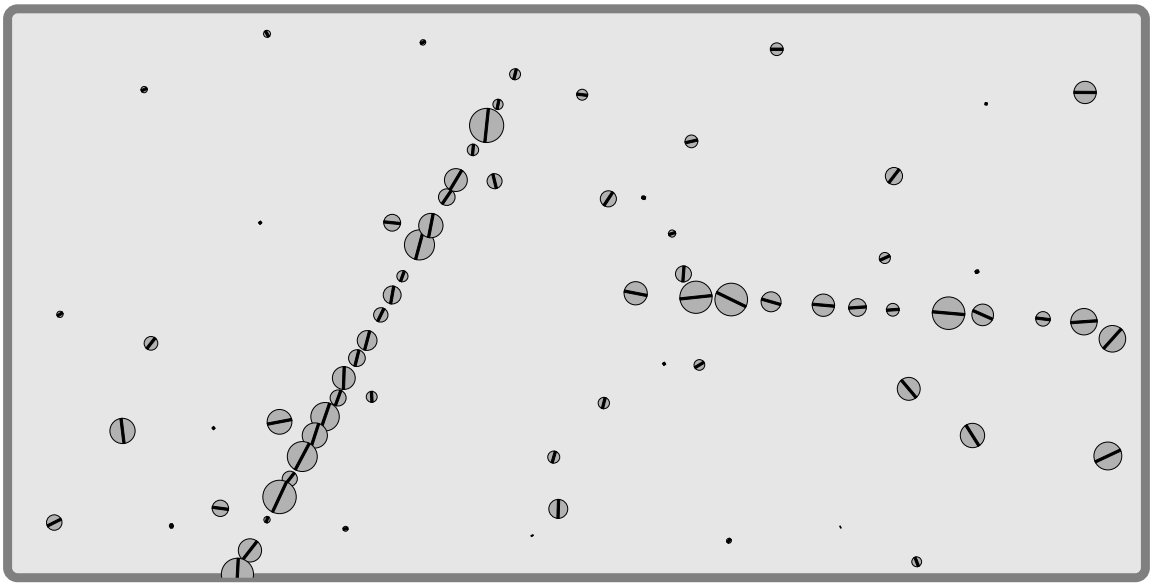


FIGURE 2.12 – Le problème offre souvent de l'information complémentaire. Ici, si un point appartient à une des deux droites recherchées, il a plus de chances d'être gros, et son orientation est *relativement* proche de celle de la droite.

de la transformation géométrique par RANSAC, cette information est complètement oubliée au profit de la seule liste des correspondances entre coordonnées obtenues précédemment.

Pourtant, si le déplacement des points d'intérêt contient l'information de la transformation, c'est aussi le cas de la déformation du signal. Or, la déformation peut être approchée simplement par les variations locales du gradient ou encore mieux, de la matrice d'autocorrélation. Ces calculs sont la plupart du temps déjà effectués pendant la recherche des caractéristiques.

En plus de la transformation, le signal peut contenir d'autres informations exploitables. Les informations de couleur et de texture sont utilisées en traitement du signal pour segmenter les objets, or, l'estimation des transformations par RANSAC est souvent proche du problème de la segmentation, notamment lorsque les objets ont des mouvements différents. Il n'y a donc aucune raison d'oublier cette information très utile.

2.4.2.2 Informations géométriques

Modèles géométriques simplifiés Il est aussi possible d'utiliser l'information apportée par des modèles géométriques simplifiés.

Exemple Nous exposons ici le compte-rendu d'une étude préliminaire qui nous a convaincu de l'utilité de modèles géométriques simplifiés.

Problème et modélisationProblème et modélisation L'entrée de notre algorithme est un ensemble $C_{i\in[1,\cdots,N_c]}$ de N_c correspondances entre des points de deux images. Chaque correspondance C_i vient de la mise en correspondance du points P_i dans la première image, avec le points P'_i dans la seconde. Les points sont exprimés en coordonnées homogènes.

Chaque paire de correspondances (C_i, C_j) définit une transformation de type similitude S_{ij} (équation 2.9 & 2.10) et une position P_{ij} (équation 2.11).

$$\begin{pmatrix} wP'k_x \\ wP'k_y \\ w \end{pmatrix} = S_{ij} \begin{pmatrix} Pk_x \\ Pk_y \\ 1 \end{pmatrix}, \quad k = i, j \tag{2.9}$$

$$S_{ij} = \begin{pmatrix} s.cos(\theta) & s.sin(\theta) & T_x \\ -s.sin(\theta) & s.cos(\theta) & T_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.10}$$

$$P_{ij} = \frac{1}{2}(Pi + Pj) \tag{2.11}$$

Appelons $T_{ij} = (S_{ij}, P_{ij})$. T_{ij} est une similitude positionnée dans le référentiel de la première image. La méthode que nous proposons approche la transformation entre les deux images avec l'ensemble des similitudes $\{T_{ij}\}_{i,j\in[1..N_c]}$. Chaque T_{ij} est placé dans l'espace à 6 dimensions $S_{xy} = (log(s), \theta, T_x, T_y, P_x, P_y)$.

Lorsque la transformation entre les deux images est une similitude, S_{ij} a la même valeur pour tous les couples de bonnes correspondances (C_i, C_j). Ainsi, les transformations T_{ij} formées avec les correspondances correctes tombent sur une variété à 2 dimensions dans S_{xy}. Il en résulte que les régions de S_{xy} correspondant aux bonnes correspondances seront denses. Au contraire, les transformations formées avec une ou deux mauvaises correspondances tomberont dans des régions éparses. Dans ce cas, on peut dire qu'un couple (C_i, C_j) est formé de deux correspondances correctes si et seulement si il mène à une région dense de S_{xy}.

Dans le cas d'un mouvement plus complexe comme une transformation affine ou projective, cette relation d'équivalence n'est plus vraie. Dans le cas particulier d'une transformation affine, les correspondances correctes avec la même position P_{ij} atteindront le même sous ensemble de S_{xy} indépendamment de la distance entre P_i et P_j. Ca justifie cette information de position jusqu'à la transformation affine. Nous avons montré que le sous ensemble atteint, forme une ellipse dans l'espace à deux dimensions $(log(s), \theta)$.

Lorsque la transformation est une projection, les deux propriétés précédentes deviennent inexactes. Une transformation définie par deux fausses correspondances peut très bien tomber sur le même point de S_{xy} qu'une transformation correcte. Cependant, nous avons constaté que la transformation similitude offre un compromis raisonnable entre la qualité de la description du mouvement et la complexité du modèle.

La figure 2.15 montre quelques formes caractéristiques dans l'espace des similitudes obtenues expérimentalement en fonction de la transformation réellement appliquée à l'image. On peut vérifier que les correspondances correctes ont, dans tous les cas, tendance à former une région dense.

Estimation d'un scoreEstimation d'un score Nous proposons d'utiliser la densité locale dans l'espace S_{xy} comme un critère de cohérence pour classer les transformations. La probabilité $P(T_{ij}\ conhérent|T_{ij})$ pour une transformation T_{ij} d'être cohérente avec les autres peut être décomposée avec la formule de Bayes :

$$P(T_{ij}\ cohérent|T_{ij}) = P(T_{ij}|T_{ij}\ cohérent)\frac{P(T_{ij}\ cohérent)}{P(T_{ij})} \tag{2.12}$$

► $P(T_{ij}|T_{ij}\ cohérent)$ est lié à la densité dans le voisinage de T_{ij}. La fonction que nous utilisons dans notre implémentation est

$$P(T_{ij}|T_{ij}\ cohérent) = \frac{densité(T_{ij})}{densité\_max}$$

Pour estimer cette densité de façon rapide, on construit un kd-tree dans S_{xy} et on approche la densité autour d'une transformation par le nombre de transformations qui partagent la même cellule du kd-tree divisé par le volume de la cellule. C'est l'opération la plus longue de notre algorithme. Sa complexité en nombre d'opérations est $O(n^2 log(n))$, où n est le nombre de correspondances à tester.

► $P(T_{ij}\ cohérent)$ peut être utilisé pour prendre en compte des informations intrinsèques. Ces informations peuvent par exemple être une échelle ou une orientation associés aux points d'intérêt. On peut

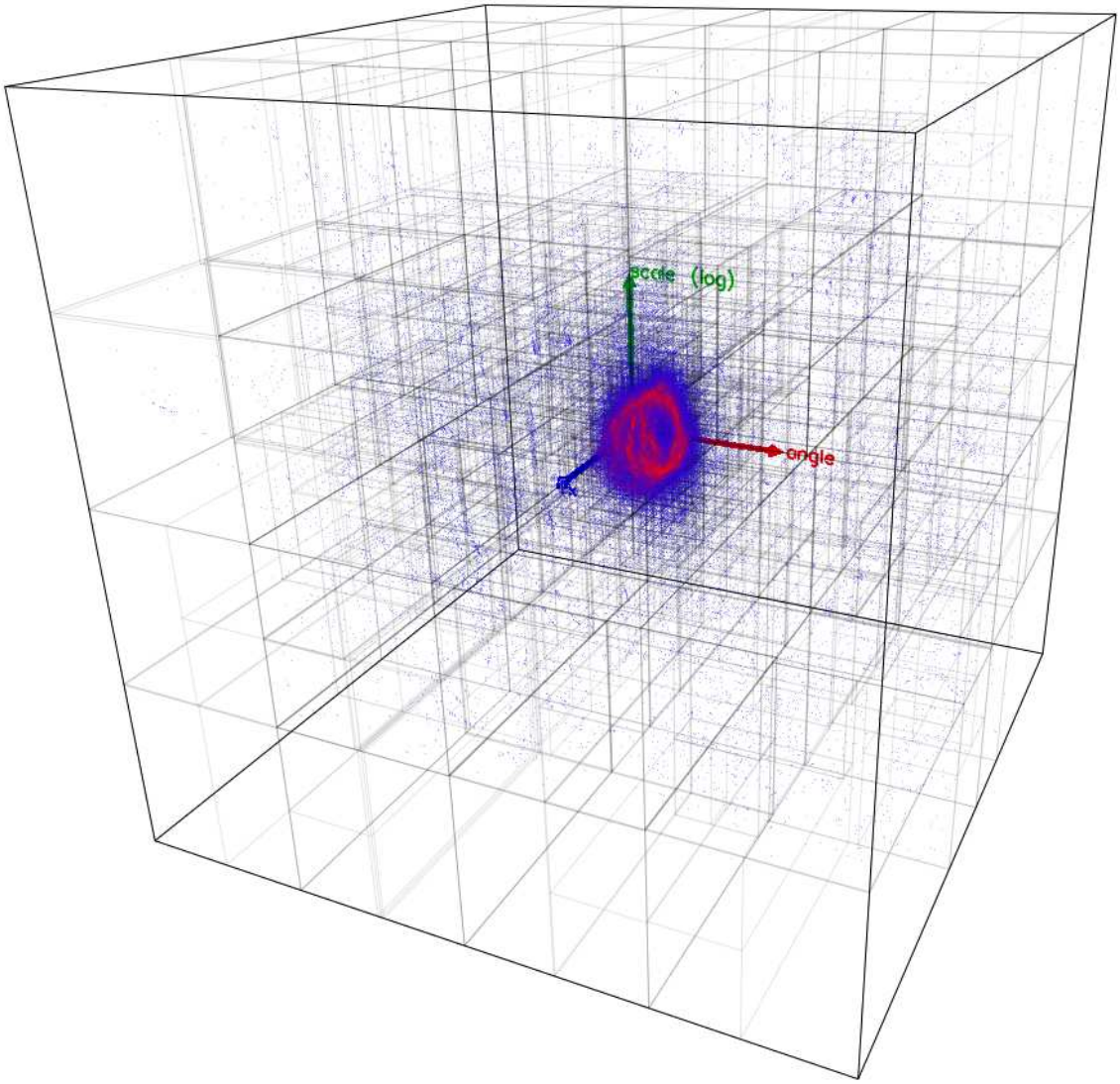


FIGURE 2.13 – Estimation rapide ($n \cdot log(n)$) de la densité de chaque région de l'espace des transformations à l'aide d'un kd-tree. Chaque point représente une transformation calculée à partir d'un couple de correspondances. Les points les plus rouges sont les transformations se trouvant dans les zones les plus denses de l'espace des transformations.

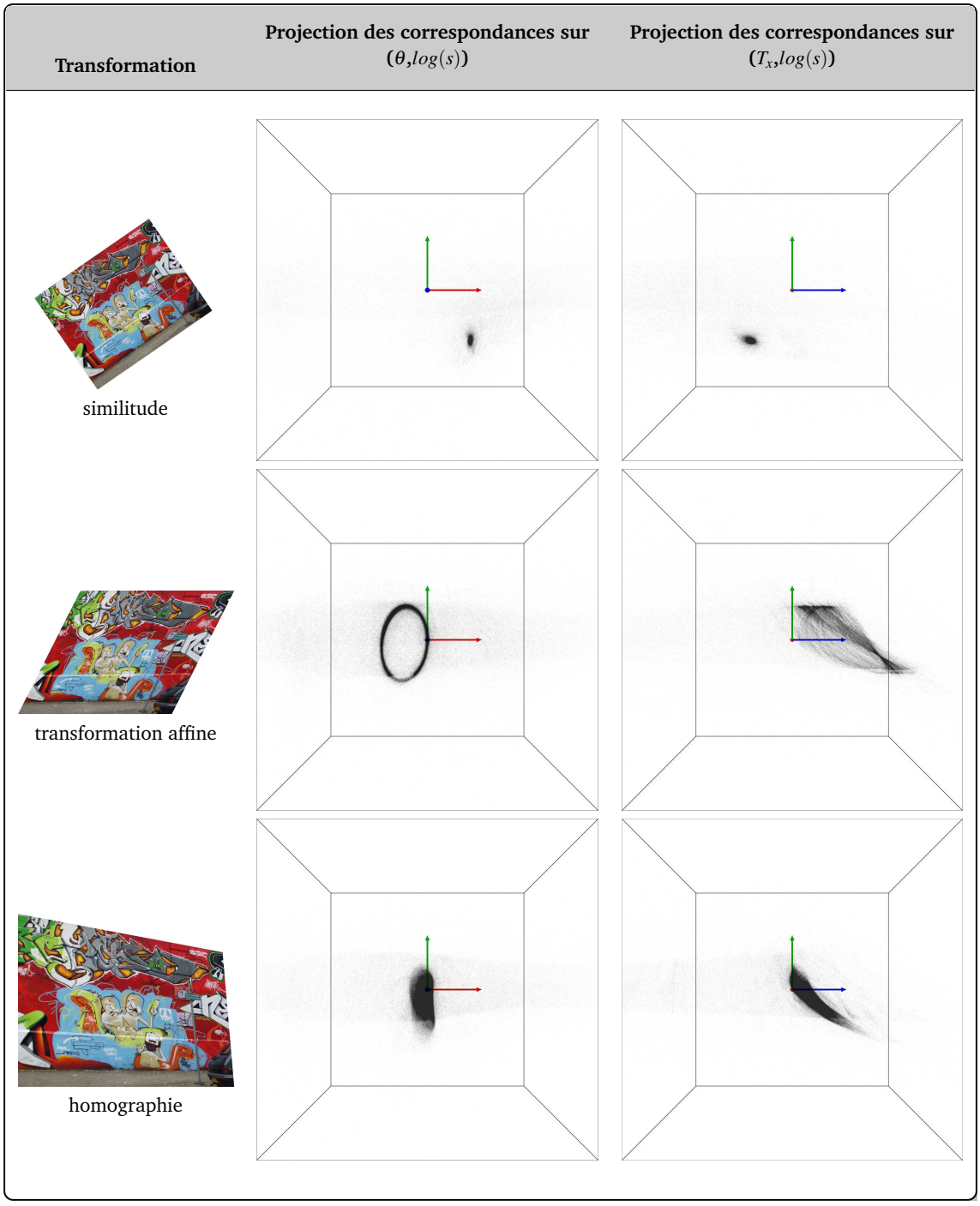


FIGURE 2.14 – Quelques exemples de formes caractéristiques obtenues dans l'espace des similitudes pour différentes transformations usuelles.

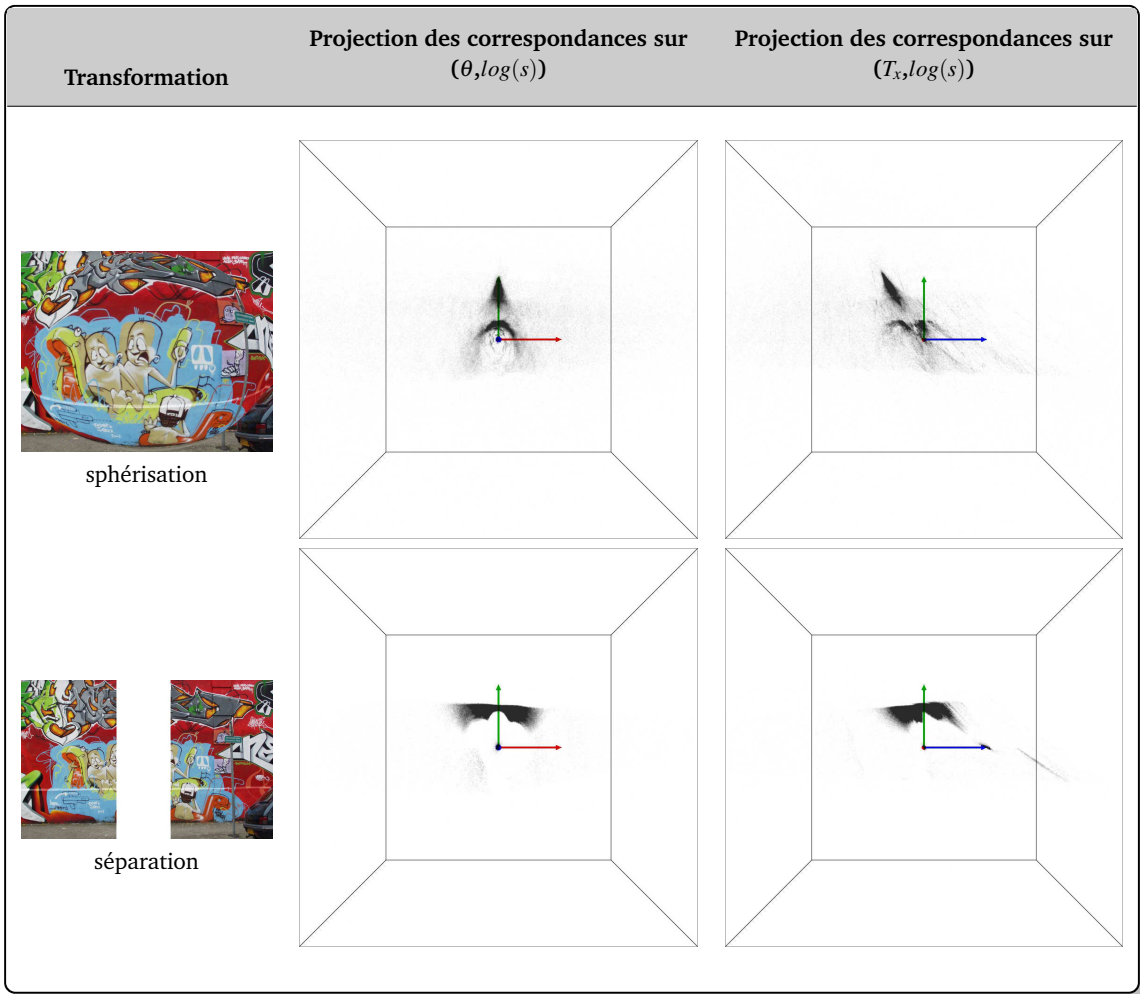


FIGURE 2.15 – Autres exemples de formes caractéristiques obtenues dans l'espace des similitudes.

aussi le faire décroître si un point d'intérêt est partagé par plusieurs correspondances. Pour le moment, nous fixons sa valeur à 1.

► Finalement, $P(T_{ij})$ peut être utilisé pour compenser le fait que les fausses transformations vont engendrer une densité non uniforme dans S_{xy}. Les raisons de cette non uniformité sont triples. La première est mathématique. Si l'on suppose une répartition uniforme des points d'intérêt dans deux images rectangulaires, et une mise en correspondance également uniforme, les densités marginales dans les six dimensions de notre espace sont toutes non-uniformes.

De plus, dans une photographie, la répartition des points d'intérêt n'est en général pas uniforme du tout quelque soit le détecteur utilisé. En effet, tout détecteur de points d'intérêt cherche à placer des points dans des zones ou il y a de l'information à capturer. Or, toutes les zones d'une image ne contiennent pas autant d'information. Les zones homogènes (ciel, texture de basse fréquence, ...) et les zones floues contiennent moins d'information que les zones très texturées et nettes. Il y a donc moins de points dans ces zones. La figure 2.16 montre un exemple de répartition des points d'intérêt.

FIGURE 2.16 – Répartition des points d'intérêt détectés avec Harris-Laplace sur une photo de scène extérieure. On voit que la répartition est loin d'être uniforme. En particulier, aucun point n'est détecté à gauche et à droite de l'arbre.

Enfin, la création de fausses mises en correspondance de points n'est pas non plus régie par une variable aléatoire de loi uniforme. Il y a deux explications à cela. D'abord, l'espace des signatures SIFT n'a pas du tout une densité homogène. Certaines signatures sont similaires à de nombreuses autres dans l'image alors que d'autres sont très distinctives. Ainsi, certains points d'intérêt peuvent être mis en correspondance avec plusieurs autres points alors que d'autres ne trouveront pas de correspondant. Cela peut-être corrigé par l'algorithme de mise en correspondance. La seconde raison pour laquelle la mise en correspondance n'est pas uniforme est que deux points sur une même texture ont plus de chances d'être mis en correspondance que deux points sur des textures différentes. Or la répartition des textures n'est bien sûr pas uniforme dans une image, ce qui biaise la variable par la position des points. Avec le descripteur SIFT cet effet n'est cependant pas très important. La figure 2.17 illustre ce biais.

Pour simplifier, nous considérons cependant que les transformations issues de correspondances erronées se répartissent uniformément dans l'espace S_{xy}. Nous fixons alors $P(T_{ij})$ à 1.

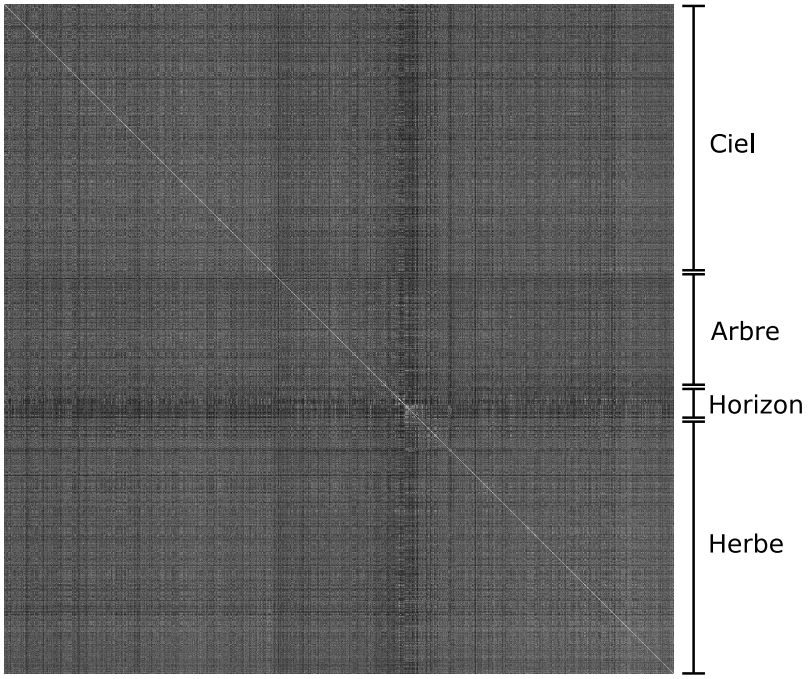


FIGURE 2.17 – Matrice de similarité des vecteurs SIFT extraits de l'image de la figure 2.16. Plus un point est clair, plus les deux vecteurs sont proches. Les points d'intérêt sont triés selon leur position verticale dans l'image. On voit clairement apparaître les différentes régions de l'image ce qui montre que la distance entre deux vecteurs SIFT est fortement liée à la position des points d'intérêt dans l'image. Les erreurs de mise en correspondance ne peuvent donc pas suivre une loi uniforme.

Une fois que l'on a estimé $P(T_{ij}\ conhérent|T_{ij})$ pour toutes les transformations T_{ij} générées, on peut en déduire la probabilité $P(C_i\ correcte)$ pour une correspondance C_i d'être correcte :

$$P(C_i\ correcte) = \frac{1}{N_c - 1} \sum_{j \neq i} P(T_{ij}\ cohérent|T_{ij}) \tag{2.13}$$

Expériences et validationExpériences et validation Cette section présente nos résultats sur quatre expériences. La première a été réalisée avec des paires d'images pour lesquelles nous possédons la vérité terrain. Pour la seconde, la vérité terrain n'est pas connue et la transformation entre les deux images peut ne pas être linéaire. La troisième expérience est une utilisation du tri obtenu pour accélérer RANSAC. Finalement, nous présentons une application de segmentation de mouvement.

Nous avons évalué notre méthode en utilisant à la fois des données réelles (photographies) et synthétiques. Les données synthétiques on été réalisées ainsi : Un ensemble de 1000 points aléatoires est généré avec une variable aléatoire uniforme. Les points image par rapport à une transformation T sont calculés et déplacés aléatoirement d'au plus deux pixels. Chacun de ces points est en suite lié soit avec sont point image, ce qui donne une correspondance correcte, soit avec un autre point, donnant une mauvaise correspondance. Pour les paires d'images réelles, on utilise les correspondances trouvées par notre implémentation du descripteur SIFT.

Transformations connues

Parce que la position des points d'intérêt est toujours sujette au bruit, on ne peut pas réellement séparer les correspondances en deux groupes, les valides et les non valides. Alors, nous présentons nos résultats sous la forme d'une courbe de l'erreur spatiale des correspondances après tri selon la probabilité estimée par notre algorithme. Les premières correspondances sont jugées les plus précises par notre

algorithme. L'erreur spatiale que nous utilisons est expliquée dans la figure 2.18.

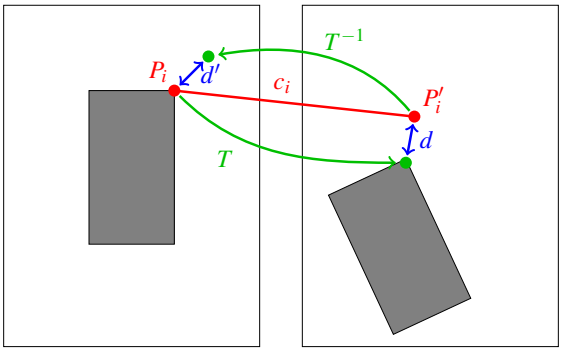


FIGURE 2.18 – L'erreur spatiale d'une correspondance C_i est définie comme l'erreur de position d' du second point par rapport au premier point et à la transformation T connue, plus l'erreur de position d du premier point par rapport au second et T^{-1}. $erreur(C_i) = d' + d = \|TP_i - P'_i\| + \|T^{-1}P'_i - P_i\|$

Nous avons testé notre algorithme sur des photographies de scènes naturelles et artificielles. La figure 2.19 est un ensemble de résultats représentatifs que nous avons obtenu.

Transformations inconnues

Quand la transformation réelle entre les deux images n'est pas connue, nous présentons le vecteur de déplacement des correspondances au dessus d'un seuil de probabilité sur l'image de gauche, et les autres correspondances (considérées comme erronées) sur celle de droite (figure 2.20). Comme notre algorithme n'est pas capable de déterminer ce seuil, il est fixé à la main pour chaque paire d'images. Les images testées présentent des changements de point de vue modérés ou des distorsions synthétiques.

Contraintes géométriques Lorsque l'on recherche la matrice fondamentale ou bien la matrice de l'homographie correspondant au mouvement d'un objet rigide, tous les ensembles de coefficients pour ces matrices ne sont pas possibles. En fait, seul un sous-ensemble des matrices fondamentales et des homographies peut être généré ainsi.

Pour la géométrie épipolaire, de nombreuses contraintes supplémentaires sont issues de la *géométrie épipolaire orientée* [27, 44, 64], qui traduit le fait qu'une caméra ne peut observer que dans une direction, devant elle.

D'une façon similaire, il a été montré que l'ensemble des homographies atteignables est un sous-espace linéaire de dimension 4, de l'espace des homographies [66], qui compte pourtant 8 dimensions. La figure 2.21 montre des exemples d'homographies atteignables et non atteignables.

2.4.2.3 Autres sources d'information

On peut imaginer de nombreuses sources d'informations complémentaires telles que le contenu des frames précédentes, l'accélération, le champ magnétique, l'image de profondeur etc.

2.4.3 Équilibre : aléatoire / déterminisme

Les méthodes basées sur l'algorithme RANSAC contiennent toutes une part d'aléatoire qui assure de ne pas tomber systématiquement dans un pire cas pour certaines données. Cependant, nombre d'entre elles apportent un peu de déterminisme dans le but d'accélérer la résolution du problème. Lors de notre étude de l'état de l'art, nous avons réalisé que de nombreuses méthodes ne sont pas homogènes en terme de randomisation (voir le tableau 2.3). Par exemple, PROSAC et GroupSAC commencent par effectuer un tri et une segmentation parfaite des données (à partir d'une information pourtant imprécise) pour ensuite

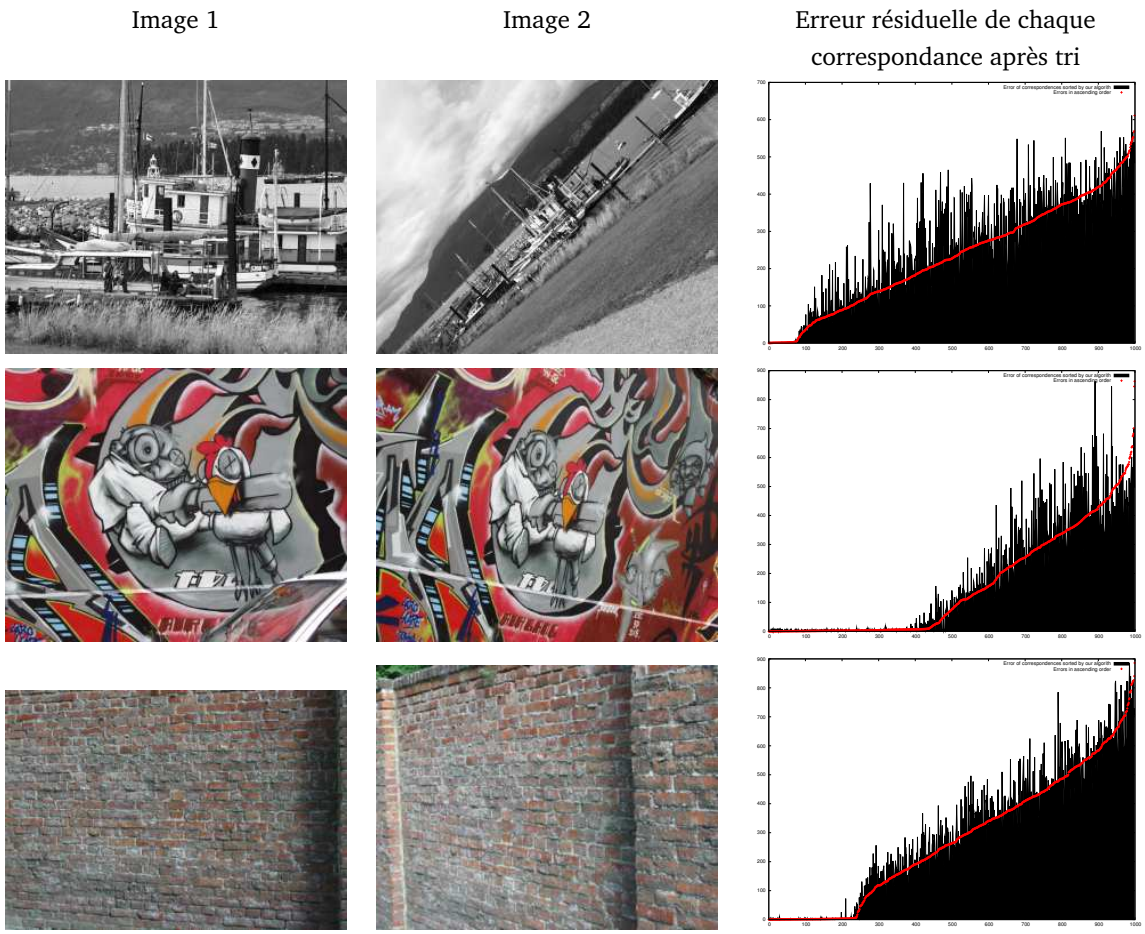


FIGURE 2.19 – Tri des correspondances pour des photographies. Le taux de bonnes correspondances varie de 8% à 50%

piocher aléatoirement dedans. RANSAC, lui-même, possède une phase de vérification d'une hypothèse qui est complète et très coûteuse, alors que les hypothèses sont générées aléatoirement et très rapidement.

Or, la probabilité de réussite de ces méthodes est liée au produit de la probabilité de réussite de chaque partie de l'algorithme. Dans ces conditions la meilleur stratégie pour maximiser cette probabilité, consiste à randomiser au même niveau chacune des parties.

Une des idées fondamentales de notre travail sera de conserver un bon équilibre randomisation/déterminisme dans les différentes parties de l'algorithme.

Méthode	Information Utilisée	Échantillonnage	Vérification Hypothèse
RANSAC	Aucune	Aléatoire	complète
RANSAC with $T_{d,d}$ test	Aucune	Aléatoire	Très icomplète
PROSAC	Imprécise	Très précis	complète
GroupSAC	Très Précises à Très imprécises	Très précis	complète
Notre objectif	Assez Précise	Assez précis	Assez précise

TABLE 2.3 – Précision accordée à chaque partie de l'algorithme par différentes méthodes. Plus le traitement d'une partie est précis, plus il est cher en temps de calcul. Trouver un bon équilibre randomisation/déterminisme est nécessaire pour concevoir une méthode efficace.

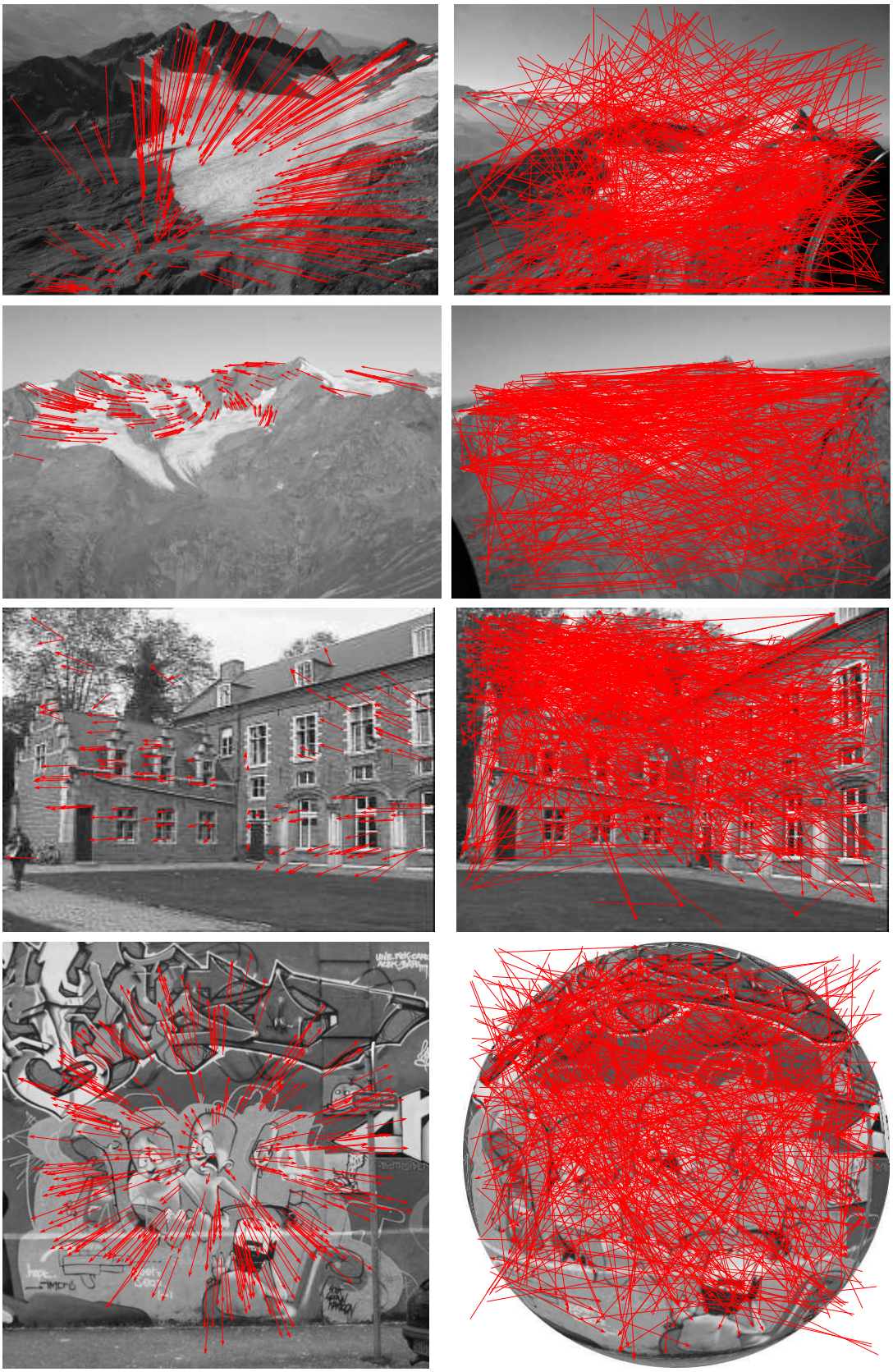


FIGURE 2.20 – Vecteurs de déplacement des correspondances considérées comme correctes (à gauches) et incorrectes (à droite). Le seuil est fixé à la main pour chaque paire d'images. Le taux de fausses correspondances varie entre 60% et 85%.

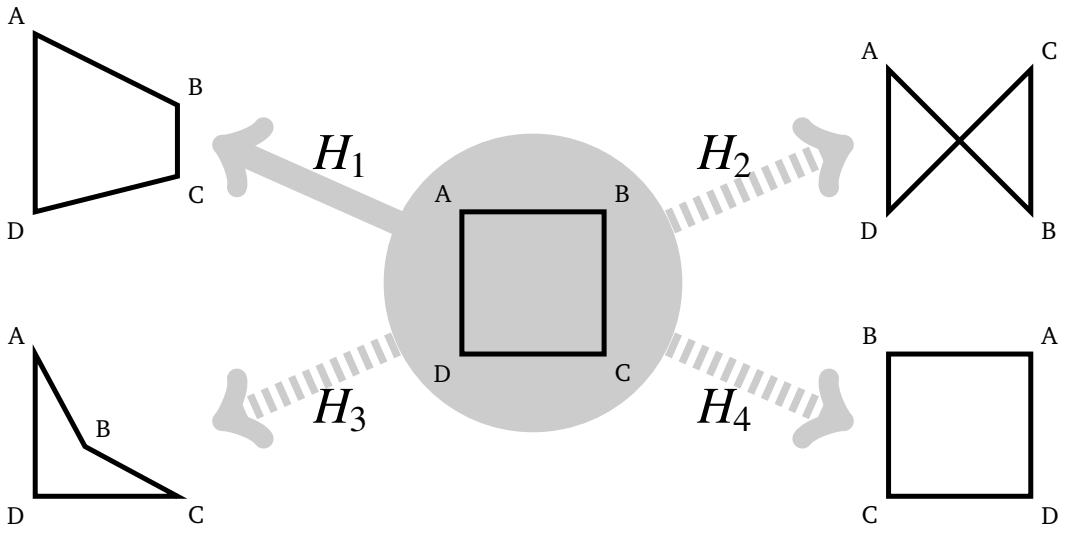


FIGURE 2.21 – Dans cet exemple, seule l'homographie H_1 est atteignable, les homographies H_2, H_3 et H_4 ne peuvent pas résulter de la projection d'un objet rigide sur deux images.

3

Unification des méthodes d'échantillonnage réordonné

Sommaire

3.1 Classification des échantillons

Dans cette thèse, nous introduisons une nouvelle approche pour la sélection d'une donnée conditionnellement au contenu de l'échantillon-hypothèse partiel. Contrairement aux méthodes précédentes, notre méthode ne devra pas supposer que la probabilité pour qu'une donnée complète avantageusement un échantillon est indépendante de cet échantillon et du contexte. Casser cette hypothèse d'indépendance permet de générer des hypothèses possédant un bien meilleur potentiel d'estimation précise des paramètres du modèle recherché.

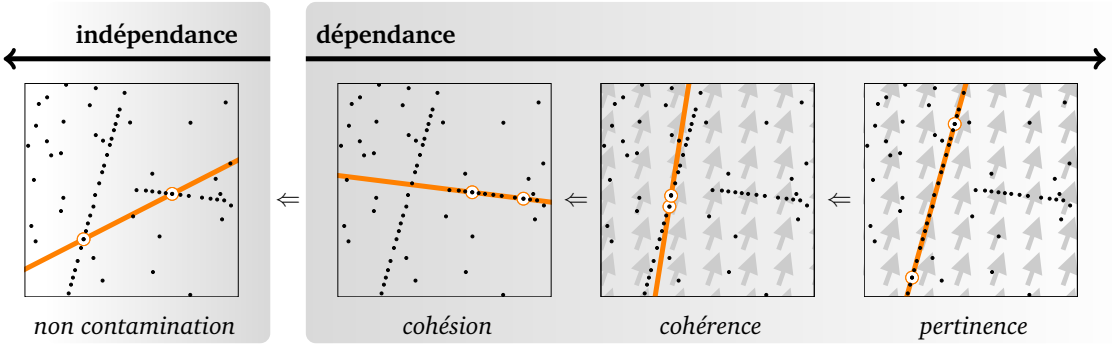


FIGURE 3.1 – Les différentes propriétés des échantillons non erronés expliquées sur le problème de l'estimation de droites.

Alors il nous faut caractériser ces échantillons ”meilleurs” que les échantillons non contaminés. Pour ce faire, nous proposons de distinguer quatre propriétés sur les échantillons : la *non contamination*, la *cohésion*, la *cohérence* et enfin la *pertinence*, telles que

$$pertinence \Rightarrow coh\acute{e}rence \Rightarrow coh\acute{e}sion \Rightarrow non\ contamination \tag{3.1}$$

Propriétés des échantillons, de la moins générale à la plus générale.

Cette section détaille chacune des propriétés proposées et la figure 3.1 en donne une explication graphique.

3.1.1 Non contamination

Les échantillons ***non contaminés*** sont les échantillons ne contenant aucune donnée aberrante. La plupart des méthodes de guidage de l'échantillonnage se concentrent sur la génération de ce type d'échantillons, ce qui a l'avantage de ne pas faire intervenir de sélection conditionnelle. C'est ce que fait PROSAC [11] en utilisant le score de mise en correspondance des points d'intérêt entre deux images. Cependant, lorsque les données contiennent plusieurs instances du modèle, un échantillon de ce type a de grandes chances de contenir des données de différentes instances, menant à une hypothèse inutile.

3.1.2 Cohésion

Les échantillons vérifiant la propriété de ***cohésion*** sont les échantillons non contaminés satisfaisant en plus des contraintes de cohérence entre eux. De nombreuses contraintes de ce type ont été étudiées dans la littérature. Nous en donnons quelques-unes dans le paragraphe 2.4.2.2. Des heuristiques ont également été utilisées comme critère de cohérence *a priori* d'un échantillon. NAPSAC (N Adjacent Points SAC) [41] utilise par exemple l'observation que les données valides sont souvent plus proches d'autres données valides que des données erronés.

3.1.3 Cohérence

Les échantillons dits ***cohérents*** sont cohérents non seulement entre eux mais avec le contexte. Dans les problèmes de vision, le contexte est une source d'information particulièrement riche. L'information contextuelle la plus utile est l'image elle-même. L'image ne fait pas partie des données en entrée de RANSAC puisqu'elle est réduite, en amont, à un simple ensemble de correspondances. Ainsi, cette information est très souvent sous-exploitée alors que comme le montre la figure 3.2, elle peut être extrêmement utile pour distinguer les bonnes données des erreurs. GroupSAC [42] montre comment une sélection dépendant d'une segmentation de l'image peut être avantageuse. Des indices sur les paramètres du modèle recherché peuvent aussi venir des frames précédentes, d'un capteur de mouvement ou d'une calibration approximative par exemple.

3.1.4 Pertinence

Les échantillons ***pertinents*** sont ceux que l'on souhaite générer. Ils sont capables de fournir une estimation précise des paramètres du modèle recherché. En particulier, il ne doivent pas être dégénérés. De nombreux critères de non dégénérescence ont été étudiés dans le cadre de l'estimation de la géométrie épipolaire. De tels critères sont généralement utilisés pour rejeter *a posteriori* les mauvais échantillons [14, 20]. Les bons échantillons sont également ceux qui sont le moins affectés par l'inévitable bruit contenu dans les données correctes. Pour cela, il est par exemple souvent préférable que les données d'un échantillon soient éloignées les unes des autres.

Notre approche doit offrir un échantillonnage conditionnel capable de générer plus d'échantillons *pertinents* que le hasard pur (i.e. uniforme) ne le ferait.

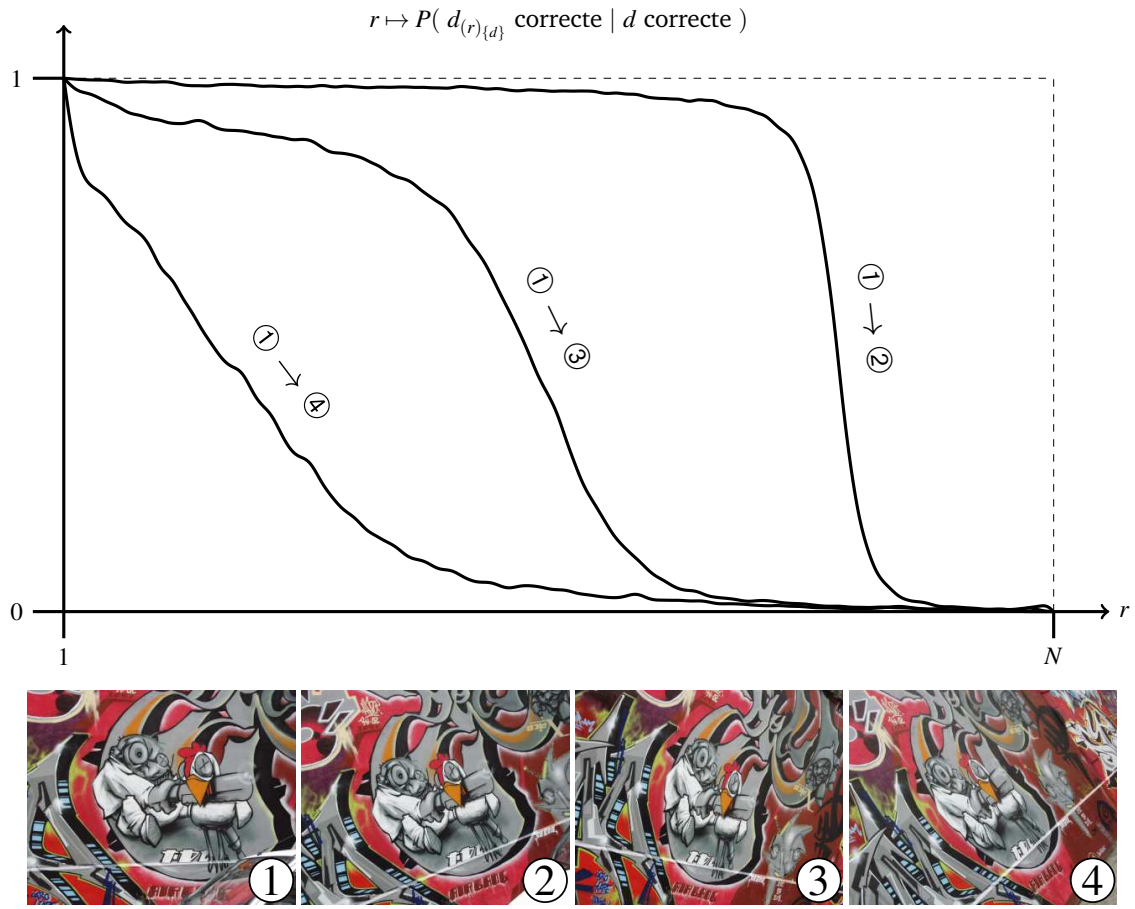


FIGURE 3.2 – Mesure de la probabilité d'être correcte de la donnée de rang r dans le tri. Ce tri est effectué en fonction d'un échantillon contenant une donnée correcte. La tâche est l'estimation d'une homographie entre deux photographies et le critère de tri est la distance entre les matrices affines associées aux correspondances. Nous utilisons le détecteur de points d'intérêt de Mikolajczyk & Schmid [35] et le descripteur SIFT [31]. Haut : valeurs de probabilités mesurées pour les comparaisons ① → ②, ① → ③ et ① → ④. Bas : images comparées. On peut remarquer que les données de rang faible ont de grandes chances d'être correctes, ce qui montre l'efficacité de ce critère de tri simple.

3.2 Une formalisation de l'échantillonnage réordonné

Les méthodes d'accélération de RANSAC par réordonnancement de l'échantillonnage (PROSAC et GroupSAC) possèdent deux propriétés qui offrent des garanties très intéressantes tout en accélérant considérablement la durée d'une estimation. Ces propriétés sont :

- $\mathscr{P}_1$: Tous les échantillons ont la même chance d'être formés au cours des T premières itérations
- $\mathscr{P}_2$: Les échantillons considérés comme meilleurs ont davantage de chances d'être formés tôt

C'est pourquoi nous avons choisi de construire notre propre algorithme, que nous appelons BetaSAC,

sur le même modèle, et nous pensons que de futurs algorithmes s'en inspireront également. Il devient donc nécessaire d'unifier ces algorithmes sous une formulation commune. C'est ce que nous nous proposons de faire dans cette section.

3.2.1 Échantillonnage

Soit $\mathscr{D} = \{d_1, \ldots, d_N\}$, un ensemble de N données et m le nombre de données nécessaires à l'estimation des paramètres du modèle recherché. Soit s un échantillon de données incomplet ($(|s| < m)$).

L'équation 3.2 définit une variable aléatoire dans $\mathscr{D}$, dépendant de l'échantillon en cours de formation s et du compteur d'itérations t. Pour $t \leq T$, la sélection se fait par une variable aléatoire $X(t,s)$. Cette variable aléatoire est ce qui caractérise chaque méthode de sélection par réordonnancement des échantillons. Puis, pour $t > T$, la sélection est uniforme, comme dans l'algorithme RANSAC original. Nous écrivons $d_{U_{\{1,\cdots,N\}}}$, la variable aléatoire retournant d_k, la donnée numéro k, où k est tiré selon la variable aléatoire uniforme $U_{\{1,\cdots,N\}}$.

On en déduit $S(t,m)$ dans l'équation 3.3, une variable aléatoire dans l'espace des échantillons de taille m. Afin d'éviter l'utilisation de lois hypergéométriques, nous faisons l'hypothèse que la sélection d'une donnée se fait toujours *avec remplacement*. Une réalisation de S peut donc contenir plusieurs fois la même donnée.

$$D(t,s) = \begin{cases} \boldsymbol{X(t,s)}, & t \leq T \\ d_{U_{\{1,\cdots,N\}}}, & t > T \end{cases} \tag{3.2}$$

Variable aléatoire de sélection d'une donnée

$$S(t,m) = \{s_1 = D(t,\{\emptyset\}), s_2 = D(t,\{s_1\}), \ldots, s_m = D(t,\{s_1,\ldots,s_{m-1}\})\ \} \tag{3.3}$$

Variable aléatoire de sélection d'un échantillon de taille m

BetaSAC et les méthodes dont il est inspiré [19, 11, 42] diffèrent tous essentiellement par leur variable aléatoire de sélection $\boldsymbol{X(t,s)}$. Cette variable est explicitée pour chacune des quatre méthode en équation 3.4.

Dans **RANSAC**, la sélection est uniforme même pour les premières itérations ($t \leq T$).

Dans **PROSAC**, nous notons $d_{(k)}$ la $k^{ième}$ donnée après tri. Ce tri est effectué une fois pour toutes en fonction d'un score de confiance associé à chaque donnée. $g(t)$ est une fonction de croissance, permettant une sélection uniforme dans l'ensemble, de plus en plus large, des données les mieux notées.

GroupSAC effectue une sélection uniforme dans une configuration définie comme une union de groupes prédéfinis de données $\mathscr{G}(t) = \{G_i\}_{i=1\ldots k}$. Comme les auteurs le démontrent, les configurations possédant peu de groupes ont plus de chances de donner des échantillons non contaminés par des données aberrantes, elle sont alors examinées plus tôt.

Pour **BetaSAC**, la variable aléatoire de sélection d'une donnée que nous proposons est une fonction de l'échantillon en cours de formation s. En cela, nous pouvons dire que BetaSAC est la première méthode à réaliser une sélection *conditionnelle* (en opposition à *indépendante*) des données d'un échantillon. La réalisation d'une variable aléatoire, $B_{i_{|s|}(t)/n}$, donne le rang de la donnée sélectionnée dans un tri dépendant de s, que nous notons $(.)_s$. Nous détaillerons dans le paragraphe 4.1.1.2 la constante n, la fonction $i_{|s|}(t)$ et la variable aléatoire $B_{i_{|s|}(t)/n}$.

$$
\begin{array}{lcl}
\bullet\ \textit{RANSAC} & \rightarrow \boldsymbol{X(t,s)} & = d_{U_{\{1,\cdots,N\}}} \\
\bullet\ \textit{PROSAC} & \rightarrow \boldsymbol{X(t,s)} & = \begin{cases} d_{(g(t))} & \text{si } |s|=0 \\ d_{(U_{\{1,\cdots,g(t)-1\}})} & \text{sinon} \end{cases} \\
\bullet\ \textit{GroupSAC} & \rightarrow \boldsymbol{X(t,s)} & = d_{U_{\mathscr{G}(t)}} \\
\bullet\ \textit{BetaSAC} & \rightarrow \boldsymbol{X(t,s)} & = d_{(B_{i_{|s|}(t)/n})_s}
\end{array}
\tag{3.4}
$$

Variable de sélection caractérisant chaque méthode d'échantillonnage réordonné

3.2.2 Objectif

Nous écrivons $\mathbb{E}$ l'espérance d'une variable aléatoire et $q(s)$ la qualité *a priori* d'un échantillon s. Nous reformulons, en équation 3.5, les propriétés $\mathscr{P}_1$ (les échantillons ont tous la même chance d'être formés) et $\mathscr{P}_2$ (ceux qui sont considérés comme meilleurs ont davantage de chance d'être formés tôt) dans le modèle mathématique proposé :

$$
\begin{array}{l}
\bullet\ \mathscr{P}_1 : \quad \mathbb{E}\left[\sum_{t=1}^{T}(S(t,m)=s)\right] = \dfrac{T}{N^m}\ , \quad \forall s \in \mathscr{D}^m \\
\bullet\ \mathscr{P}_2 : \quad t \leq t' \ \Rightarrow\ \mathbb{E}\left[q(S(t,m))\right] \geq \mathbb{E}\left[q(S(t',m))\right]
\end{array}
\tag{3.5}
$$

Propriétés devant être respectées

La fonction de qualité q de chacune des méthodes est donnée en équation 3.6.

Dans **RANSAC**, cette fonction a la même valeur pour tous les échantillons. *A priori*, RANSAC considère que les échantillons ont tous la même chance d'estimer les paramètres du modèle recherché.

PROSAC considère qu'un échantillon est pénalisé par la donnée de moindre qualité de celui-ci. C'est pourquoi pour lui, la qualité d'un échantillon vaut la qualité q_{PROSAC} associée à la moins bonne de ses données. En pratique, q_{PROSAC} est liée à la confiance de la mise en correspondance des deux points d'intérêts.

GroupSAC repose sur l'hypothèse qu'il n'est pas prudent de mélanger des données issues de groupes différents. Ainsi, la fonction de qualité sous-jacente à l'algorithme est l'inverse du nombre de groupes différents représentés dans l'échantillon s.

La fonction $q_{BetaSAC(s)}$ utilisée dans **BetaSAC** est détaillée dans la section 4.1.1.2.

$$
\begin{array}{lcl}
\bullet\ \textit{RANSAC} & \rightarrow \boldsymbol{q(s)} & = 1 \\
\bullet\ \textit{PROSAC} & \rightarrow \boldsymbol{q(s)} & = \min\left\{q_{PROSAC}(s_k)\right\}_{k\in\{1,\cdots,m\}} \\
\bullet\ \textit{GroupSAC} & \rightarrow \boldsymbol{q(s)} & = \text{Inverse du nombre de groupes représentés dans } s \\
\bullet\ \textit{BetaSAC} & \rightarrow \boldsymbol{q(s)} & = q_{BetaSAC(s)}
\end{array}
\tag{3.6}
$$

Fonction de qualité *a priori* d'un échantillon s utilisée dans chaque méthode

4

Nouvelles méthodes d'échantillonnage : BetaSAC et OABSAC

Sommaire

4.1 BetaSAC

4.1.1 Nouvelle variable aléatoire de sélection

BetaSAC est caractérisé par sa variable aléatoire de sélection, $X_{BetaSAC}(t,s)$, définie dans l'équation 3.4, où s est l'échantillon en cours de création à l'itération t. $X_{BetaSAC}(t,s)$ est le résultat de la sélection de la $k^{ième}$ donnée dans un tri dépendant de s, où k est un nombre dans $\{1,\dots,N\}$ tiré par la variable aléatoire $B_{i_l(t)/n}$. Dans la section précédente, nous avons présenté des exemples de critères utilisables pour définir un classement des données en fonction d'un échantillon partiel s. Dans cette section, nous définissons la variable aléatoire $B_{i_l(t)/n}$.

$B_{i_l(t)/n}$ doit permettre de contrôler la région du tri dans laquelle une donnée est tirée. Au cours des premières itérations, les données les mieux classées dans le tri doivent être sélectionnées, tout en satisfaisant la propriété $\mathscr{P}_1$ qui assure que chaque échantillon a eu la même chance d'être formé après T itérations. Trouver une telle variable aléatoire est facile. Nous avons retenu une sous-famille de la loi bêta. La loi bêta de paramètres (α,β) est définie par la fonction de densité de l'équation 4.1, avec $\text{Beta}(\alpha,\beta)$, la fonction bêta définie dans l'équation 4.2.

$$f_{Beta}(x;\ \alpha,\ \beta) = \frac{1}{\text{Beta}(\alpha,\beta)}\, x^{\alpha-1}(1-x)^{\beta-1} \tag{4.1}$$

$$\text{Beta}(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\, dt \tag{4.2}$$

La fonction de densité $f_{B_{i_l(t)/n}}$ de notre variable aléatoire est définie à partir de la loi bêta dans l'équation 4.3, et dessinée dans la figure 4.1.

$$f_{B_{i_l(t)/n}}(x) = \frac{1}{N-1} f_{Beta}\left(\frac{x-1}{N-1}\ ;\ i_l(t)\ ,\ n-i_l(t)+1\right) \tag{4.3}$$

Fonction de densité caractéristique de la sélection des données dans BetaSAC

Le paramètre n est une constante dans $\mathbb{N}^*$ et la suite de vecteurs $\vec{i}(t) = [i_0(t),\cdots,i_{m-1}(t)]$ se déplace dans $\{1,\dots,n\}^m$ au cours des itérations. Nous discuterons de leurs valeurs dans la section 4.1.1.2.

Les raisons pour lesquelles nous choisissons la loi bêta sont multiples. La plus importante est qu'elle ne requière pas un tri complet des données. Grâce à cette propriété, le coût additionnel de notre procédure d'échantillonnage est négligeable. De plus, elle est triviale à simuler et permet de satisfaire $\mathscr{P}_1$ et $\mathscr{P}_2$ facilement.

4.1.1.1 Simulation de la variable aléatoire.

La loi bêta est la statistique d'ordre de la loi uniforme, ce qui la rend triviale à simuler à partir d'un générateur aléatoire uniforme. Étant donné un compteur d'itérations t, un tirage aléatoire complet d'un échantillon s et taille m par notre variable aléatoire conditionnelle est présenté dans l'algorithme 1.

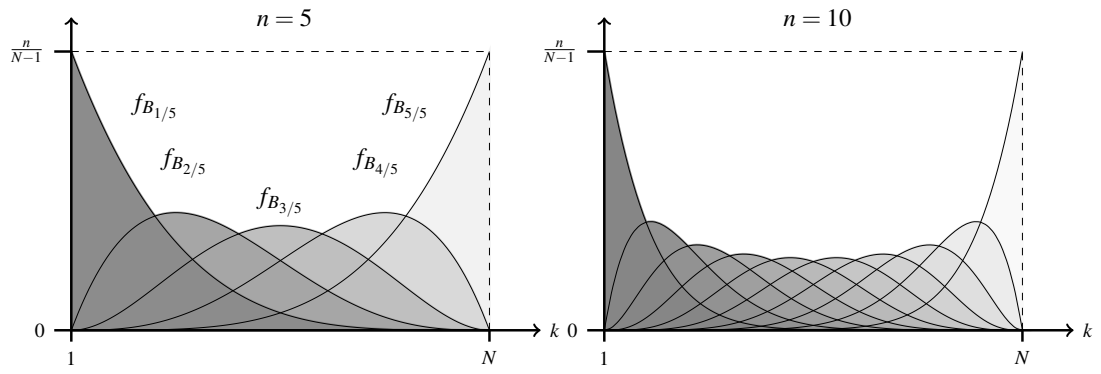


FIGURE 4.1 – Fonctions de densité des n variables aléatoires $B_{i/n}, i \in \{1,\cdots,n\}$ pour $n = 5$ (gauche) et $n = 10$ (droite). Les densités les plus sombres correspondent aux plus petites valeurs de i. Avec un bon critère de tri, les premières données ont plus de chances d'être correctes. Donc, $B_{i/n}$ sélectionne plus de données correctes que le hasard pur quand i est petit.

Algorithm 1: Tirage d'un échantillon complet par notre variable aléatoire $S(t,m)$

Data: t : compteur d'itérations
Data: m : taille d'un échantillon-hypothèse
$s \leftarrow \{\emptyset\}$
for $l = 0$ *to* $m-1$ **do**
  Tirer n données selon le hasard uniforme
  Trier les n données en fonction de s
  Trouver d, la $i_l(t)^{ième}$ donnée parmi les n tirées précédemment
  $s \leftarrow s \cup d$

En remplaçant le tri complet des n données par un algorithme de sélection linéaire, la complexité en temps de calcul de cette procédure d'échantillonnage est seulement $O(m.n)$.

4.1.1.2 Stratégie d'échantillonnage

Chaque itération de BetaSAC commence par le calcul de ce que nous appellerons le *vecteur de sélection* $\vec{i}(t) = [i_0(t),\ldots,i_{m-1}(t)]$, pour l'itération courante t. Nous appelons cette suite de vecteurs la *stratégie d'échantillonnage*. Dans cette section, nous définissons une stratégie d'échantillonnage qui satisfait les propriétés $\mathscr{P}_1$ et $\mathscr{P}_2$. Comme la randomisation est assurée par $B_{i_l(t)/n}$ même pour des valeurs $i_l(t)$ déterministes, il n'est pas nécessaire que $i_l(t)$ soit une variable aléatoire. Cependant, nous considérons le cas général d'une variable aléatoire discrète dans $\{1,\cdots,n\}$.

À partir de l'identité suivante,

$$\frac{1}{n}\sum_{i=1}^{n} f_{B_{i/n}} = f_{U_{[1,N]}} \tag{4.4}$$

nous pouvons déduire deux propriétés suffisantes *(a)* et *(b)* sur $i_l(t)$ pour satisfaire $\mathscr{P}_1$:

$$\forall (j,j') \in \{1,\cdots,n\}^2,\ \ \forall (l,l') \in \{0,\cdots,m-1\}^2,$$

$$(a) \qquad \sum_{t=1}^{T} P(i_l(t) = j) = \frac{T}{n} \tag{4.5}$$

$$(b) \quad l \neq l' \Rightarrow \sum_{t=1}^{T} P(i_l(t) = j)).P(i_{l'}(t) = j')) = \frac{T}{n^2}$$

La propriété $\mathscr{P}_2$ fait intervenir une fonction de qualité d'un échantillon, q. Il n'existe pas de fonction q optimale sans hypothèse forte sur le tri $(.)_s$. Cependant, nous présentons une fonction $q_{BetaSAC}$ qui est une généralisation naturelle de la qualité *a priori* d'un échantillon utilisée dans PROSAC. Soit $\vec{r}_s = [r_1, \ldots, r_m]$ le vecteur des rangs des données sélectionnées dans un échantillon s de taille m ($\vec{r}_s$ est le vecteur des m réalisations des variables aléatoires $B_{i_0(t)/n}, \ldots, B_{i_{m-1}(t)/n}$). Autrement dit :

$$s = \{s_1 = d_{(r_1)_{\{\emptyset\}}}, s_2 = d_{(r_2)_{\{s_1\}}}, \ldots, s_m = d_{(r_m)_{\{s_1,\ldots,s_{m-1}\}}}\}$$

La fonction de qualité *a priori* d'un échantillon s utilisée dans BetaSAC est définie comme l'opposé de la *p-norme* du vecteur des rangs :

$$q_{BetaSAC}(s) = -\|\vec{r}_s\|_p = -\sqrt[p]{\sum_{l=1}^{m} r_l^p} \tag{4.6}$$

Fonction de qualité *a priori* d'un échantillon s utilisée dans BetaSAC

Ce critère de qualité à l'avantage de donner une forme fermée de la propriété $\mathscr{P}_2$ (équation 4.7, où le symbole $f(t) \sim g(t)$ indique l'existence d'une fonction croissante h telle que $f = h \circ g$).

$$\begin{aligned}\mathbb{E}\left[q_{BetaSAC}(S(t,m))\right] &\sim -\sum_{l=0}^{m-1} \mathbb{E}\left[B^p_{i_l(t)/n}\right] \\ &\sim \underbrace{-\sum_{l=0}^{m-1}\sum_{j=1}^{n} P(i_l(t)=j)\prod_{k=1}^{p} j+k-1}_{\overset{\text{déf.}}{=} E_p\left(\{i_l(t)\}_{l=0\ldots m-1}\right)}\end{aligned} \tag{4.7}$$

$$\mathscr{P}_2 : t \leq t' \Rightarrow E_p\left(\{i_l(t)\}_{l=0\ldots m-1}\right) \leq E_p\left(\{i_l(t')\}_{l=0\ldots m-1}\right)$$

La valeur optimale pour p dépend de la qualité du tri. Si on suppose le tri parfait (les données valides occupent les premières positions), le meilleur choix est celui qui minimise le rang maximum dans l'échantillon, c'est-à-dire $p = \infty$. Mais dans le cas d'un tri très imparfait, une faible valeur de p est un meilleur choix. Nos expérimentations nous ont montré que $p = 3$ est souvent la valeur optimale. Dans les cas où elle ne l'est pas, elle reste cependant une bonne valeur. C'est donc celle que nous avons choisi dans notre implémentation. Notez que pour $(n,p) = (\infty,\infty)$, notre échantillonnage est parfaitement équivalent à celui de PROSAC. C'est possible en pratique, mais choisir $n = \infty$ demande de trier toutes les données à chaque sélection, puisque notre tri peut dépendre de l'état courant de l'échantillon s.

Maintenant que nous avons une formulation simple de $\mathscr{P}_1$ et $\mathscr{P}_2$ (équation 4.5 et 4.7), nous pouvons définir la fonction $i_l(t)$ utilisée. Soient $u_1(t), \ldots, u_m(t)$ les m chiffres en base n du nombre $\lfloor \frac{t}{T} n^m \rfloor$. $i_l(t)$, définie ainsi :

$$i_l(t) = u_l(f(t)) + 1, \quad \forall l \in \{0, \cdots, m-1\} \tag{4.8}$$

satisfait les propriétés (a) et (b) de l'équation 4.5, pour n'importe quelle permutation f de $\{1, \cdots, T\}$. La permutation f doit être choisie de façon à satisfaire $\mathscr{P}_2$. f peut être précalculée par n'importe quel algorithme de tri ou bien calculée en ligne en utilisant la méthode "Fast Marching" (FMM) comme décrit en figure 4.3.

La FMM est appliquée à un arbre des vecteurs de stratégie construit de telle façon que que chaque nœud fils possède le vecteur de stratégie de son nœud père, dont une valeur est incrémentée.
De cette manière, quelque soit la fonction de qualité utilisée, on est certain que tous les nœuds fils ont une priorité inférieure à leur nœud père.

Valeur de *n*. Il est important de noter que n n'a pas à être défini en fonction du nombre total de données

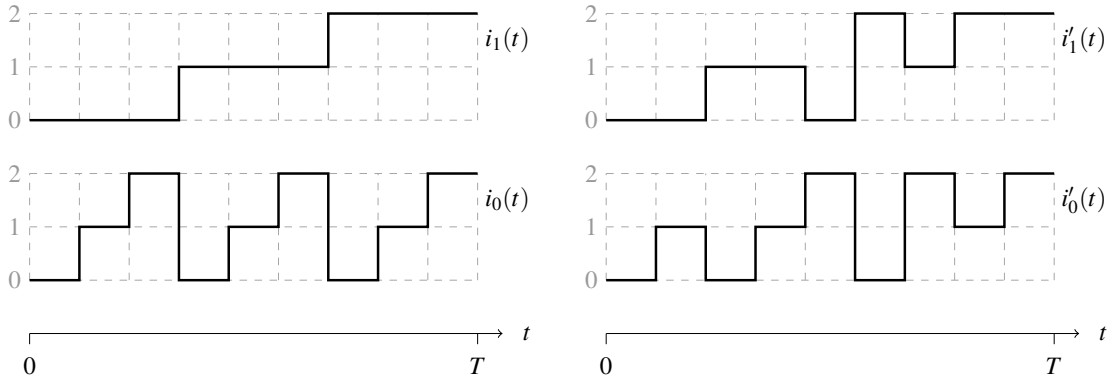


FIGURE 4.2 – Exemples d'ordonnancement des vecteurs de sélection avec $m = 2$. Les vecteurs $[i'_0(t), i'_1(t)]$ (à droite) sont mieux ordonnés que les vecteur $[i_0(t), i_1(t)]$ définis par l'équation 4.8 (à gauche)

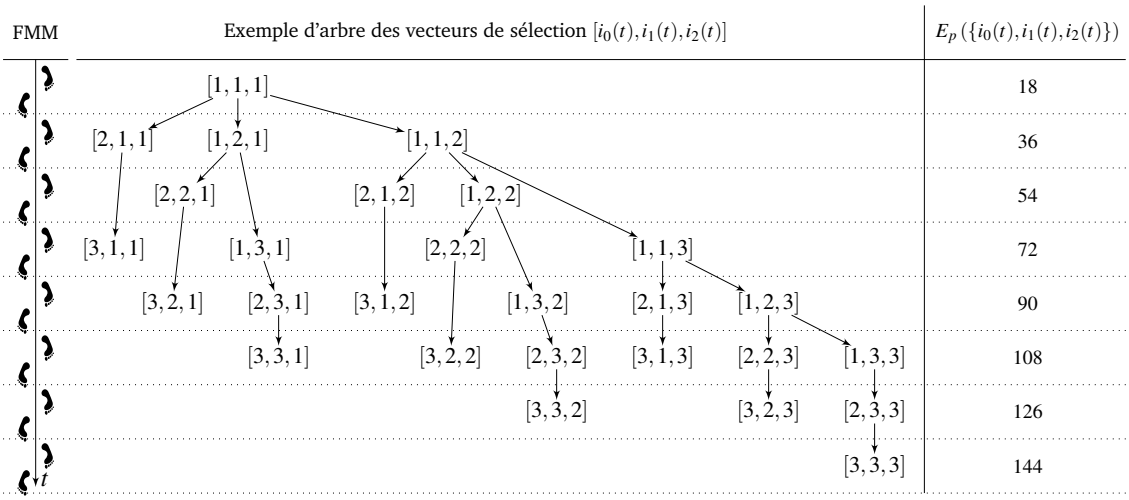


FIGURE 4.3 – Ordonnancement des vecteurs de sélection $[i_0(t), \ldots, i_{m-1}(t)]$ en fonction de $E_p(\{i_0(t), \ldots, i_{m-1}(t)\})$ par la méthode Fast Marching (FMM), pour $m = 3$, $n = 3$ et $p = 3$.

N. Sa valeur optimale dépend de la difficulté du problème que l'on cherche à résoudre (principalement le taux de données valides et le nombre de modèles coexistant).

Soit $\sigma_{B_{i/n}}$ l'écart type de $B_{i/n}$ (équation 4.9). La valeur de $\sigma_{B_{i/n}}$ détermine la localité de la sélection d'une donnée dans la liste triée. Elle dépend directement de n. Un problème difficile requière une sélection très locale car seulement les toutes premières données du tri doivent être sélectionnées dans les premières itérations.

$$\sigma_{B_{i/n}} = \frac{1}{n+1}\sqrt{\frac{i(n-i+1)}{n+2}} \tag{4.9}$$

Pour toute valeur de i fixée, $\sigma_{B_{i/n}}$ est asymptotiquement équivalent à $\frac{\sqrt{i}}{n}$ quand n devient grand. Ainsi, n doit être fixé proportionnellement à l'inverse du taux de données susceptibles de compléter un échantillon de façon pertinente. Pour la plupart des applications en vision par ordinateur, $n = 10$ est une valeur suffisante. C'est celle que nous utiliserons dans la partie expérimentale.

4.2 OABSAC : optimal and adaptative BetaSAC

Le but poursuivi pendant la conception de OABSAC est de créer une méthode optimale - d'un certain point de vue que nous expliciterons - proche de BetaSAC. Pour ce faire, il nous a fallu :

1. Formaliser la **notion de pertinence** d'un échantillon-hypothèse.
2. Concevoir un **apprentissage *hors ligne*** de la stratégie optimale pour générer des échantillons de pertinence maximale pendant les premières itérations.
3. Concevoir la **partie *en ligne*** de la méthode, capable d'utiliser le résultat de l'apprentissage et éventuellement de s'adapter aux données au cours des itérations.

Dans cette section, nous justifions la nécessité d'une telle méthode en remarquant les limites de BetaSAC, puis nous détaillons notre travail sur les trois points mentionnés précédemment.

4.2.1 Limites de BetaSAC

BetaSAC est très simple à implémenter et très performant. Cependant, certaines contraintes sont superflues et certains paramètres sont fixés arbitrairement. En les relâchant, on peut espérer gagner encore en performance.

4.2.1.1 Contraintes inutiles

Dans BetaSAC, la valeur du paramètre n (voir l'algorithme 1) est la même pour chaque étape de la création d'un échantillon. Il n'y a pas de raison qu'il ne dépende pas de la taille courante de l'échantillon d'autant que les informations utilisées pour chaque taille l sont potentiellement différentes. Par exemple, les indices de cohérence n'ont du sens que lorsque l'échantillon contient au moins un élément (i.e. $l > 1$), pas avant.

Dans OABSAC, la variable n sera alors remplacée par un vecteur $\vec{n}$ de dimension m.

4.2.1.2 Paramètres fixés arbitrairement

Les paramètres n et p de BetaSAC ont été fixés expérimentalement de façon à obtenir de bons résultats sur nos données de test. Mais, pour une application autre que celles que nous considérons dans cette thèse, ou bien pour des données très différentes il se peut que d'autres valeurs conviennent mieux.

Alors, le vecteur $\vec{n}$ optimal ainsi que le critère d'ordonnancement des vecteurs $\vec{i}$ utilisés dans OABSAC devront être définis dans la phase d'apprentissage.

4.2.1.3 Difficulté de mélanger des informations de types différents

Il est courant, en mathématiques appliquées, de vouloir utiliser au mieux des informations de types différents. Cela nécessite d'être capable de les comparer entre elles ce qui n'est souvent pas possible dans le cas général. Dans notre travail, on aimerait ainsi pouvoir utiliser le maximum d'information disponible pour ordonner les échantillons candidats. L'importance que l'on doit donner à chaque information dépend du problème à résoudre mais aussi des données que l'on va rencontrer. Là encore, seule une phase d'apprentissage sur un ensemble de données représentatives, peut nous aider.

4.2.1.4 Nécessité d'une adaptation en ligne

La stratégie de BetaSAC (i.e. l'ordre de la suite des vecteurs de sélection $\vec{i}(t) = [i_0(t), \cdots, i_{m-1}(t)]$) optimale dépend du problème que l'on cherche à résoudre. Précisément, elle dépend des fonctions de densité $r \mapsto f_{P\left(d_{(r)_s} \text{ pertinent } \mid s\right)}$. Les fonctions de densité moyennes peuvent être obtenues par l'apprentissage, mais nous avons souhaité que OABSAC soit également capable de s'adapter aux différences entre les données traitées et les données utilisées lors de la phase d'apprentissage. La figure 4.4 permet de se convaincre

que le taux de données correctes I, à lui seul est un paramètre à prendre en compte. Or, comme dans RANSAC, l'estimation de la valeur du taux I varie aux cours des itérations. C'est pourquoi il est nécessaire d'adapter la stratégie pendant les itérations de OABSAC.

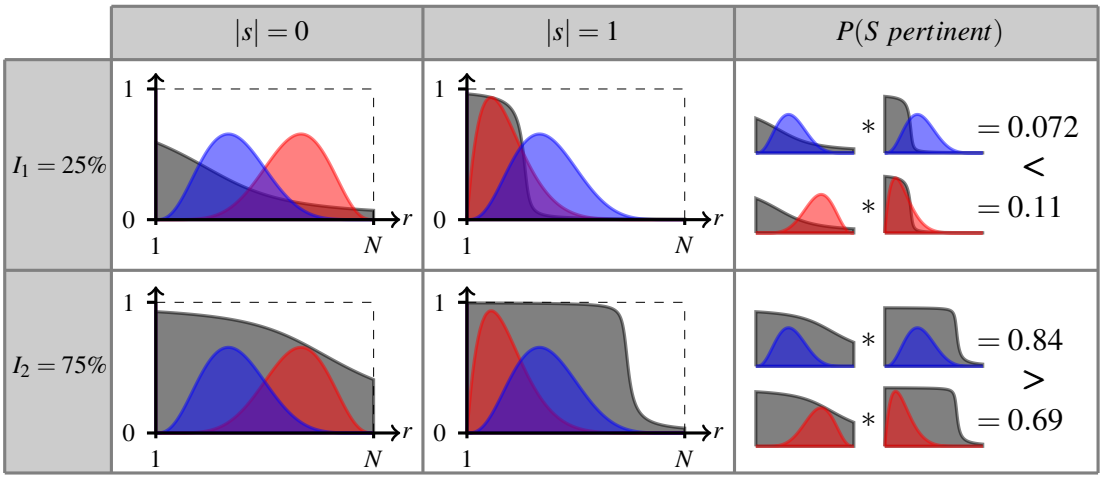


FIGURE 4.4 – Probabilité, $P(S\ pertinent)$, de tirer un échantillon complet pertinent, pour deux taux de données correctes, I_1 et I_2, et deux vecteurs de sélection différents. Le vecteur de sélection rouge est meilleur pour $I = I_1 = 25\%$ alors que pour $I = I_2 = 75\%$, le bleu est meilleur. gris : $r \mapsto P(d_{(r)_s}\ pertinent)$, bleu : fonction de densité de $B_{v_1(l)/n}$, rouge : fonction de densité de $B_{v_2(l)/n}$. Cette figure est issue d'un calcul théorique dont les détails sont donnés dans le paragraphe 4.2.3.2.

4.2.2 Pertinence d'un échantillon

Dans la section 3.1, nous montrons à quel point les notions d'échantillon *non contaminé* et d'échantillon *pertinent* sont éloignées. Les premiers sont, dans le cas général, incapables de fournir une estimation, même approximative, des paramètres du modèle recherché. Mais jusqu'à présent, nous n'avons pas donné de définition formelle de la pertinence d'un échantillon.

4.2.2.1 Définition de la pertinence d'un échantillon

Nous définissons la notion de pertinence d'un échantillon s, complet ou non (i.e. $|s| \leq m$), par l'équation 4.10, où U représente la variable aléatoire uniforme dans l'ensemble des données $\mathscr{D}$.

$$pertinence(s) = \begin{cases} \mathbb{E}\left[pertinence(s \cup \underbrace{U \cup \cdots \cup U}_{m-|s|})\right] \Big/ \left(\mathbb{E}\left[pertinence(\underbrace{U \cup \cdots \cup U}_{m})\right]\right)^{m-|s|} & \text{si } |s| < m \\ P(s \rightarrow \text{bonne approximation du modèle recherché}) & \text{si } |s| = m \end{cases}$$

Définition de la *pertinence* d'un échantillon s

(4.10)

La probabilité $P(s \rightarrow$ bonne approximation du modèle recherché$)$, que l'échantillon s instancie un bon modèle est l'**unique paramètre à définir par l'utilisateur de OABSAC**. Sa définition pratique n'est utilisée que pendant la phase d'apprentissage, elle peut donc être définie à partir de la vérité terrain. Dans nos expériences, nous avons utilisé simplement

$$P(s \rightarrow \text{ bonne approximation du modèle recherché}) = \begin{cases} 0 & \text{si } score(s) < \alpha \cdot I \cdot N \\ 1 & \text{sinon} \end{cases} \tag{4.11}$$

avec $\alpha = 0.7$.

4.2.2.2 Propriétés de la pertinence d'un échantillon

La pertinence d'un échantillon, telle qu'elle est définie par les équations 4.10 et 4.11, possède les deux propriétés ci-dessous, pour tout échantillon s :

$$\begin{aligned} &\textbf{1.} \quad pertinence(s) \in \begin{cases} \{0\} & \text{si } s \text{ est contaminé, incohérent ou dégénéré} \\ [0,1] & \text{sinon} \end{cases} \\ &\textbf{2.} \quad \mathbb{E}\left[pertinence(\underbrace{U \cup \cdots \cup U}_{m}) \right] \leq I^m \end{aligned} \tag{4.12}$$

La pertinence est donc une propriété beaucoup plus forte que la non contamination.

4.2.2.3 Exemples

Nous proposons quelques exemples simples de valeurs de pertinence mesurées, destinés à montrer à quel point cette notion est très contextuelle, dans le sens où elle dépend fortement des autres données de l'échantillon.

Cercles La figure 4.5 montre un exemple de la valeur de la pertinence des données dans le cas d'un problème de détection de cercles dans un nuage de points 2D. Dans ce cas, la taille échantillons-hypothèse, m, vaut 3.

Homographies La figure 4.6 présente un exemple de la valeur de la pertinence des données dans le cas d'un problème d'estimation de l'homographie entre deux images ($m = 4$).

Géométrie épipolaire La figure 4.7 présente un exemple de la valeur de la pertinence des données dans le cas d'un problème d'estimation de la géométrie épipolaire entre deux images ($m = 7$).

Les tableaux 5.1 et 5.2 présentés dans la partie 5 donnent des exemples de courbes de pertinence dans le contexte de l'estimation d'homographies et de matrices fondamentales.

4.2.3 Description de OABSAC

OABSAC est composé d'une partie hors ligne, d'apprentissage, et une partie en ligne très similaire à celle de BetaSAC possédant éventuellement une procédure d'adaptation au cours des itérations en plus.

Dans ce chapitre, on appellera *mesure*, toute quantité que l'on peut mesurer sur un échantillon de données de taille quelconque. Le but de OABSAC, est d'utiliser un ensemble de mesures pour former des échantillons les plus pertinents possible le plus tôt possible. Des exemples de mesures sont donnés dans le tableau 5.3.

L'apprentissage est indispensable pour combiner au mieux des mesures de types différents, pour un problème donné. La partie hors ligne se compose de trois étapes.

– $\mathbf{A}_1$) Apprentissage de l'association entre le résultat d'un ensemble de mesures à une valeur de pertinence. Il en résulte une fonction pour chaque taille d'échantillon $|s| = l$, appelée $\boxed{f^l_{m \rightarrow p}}$.

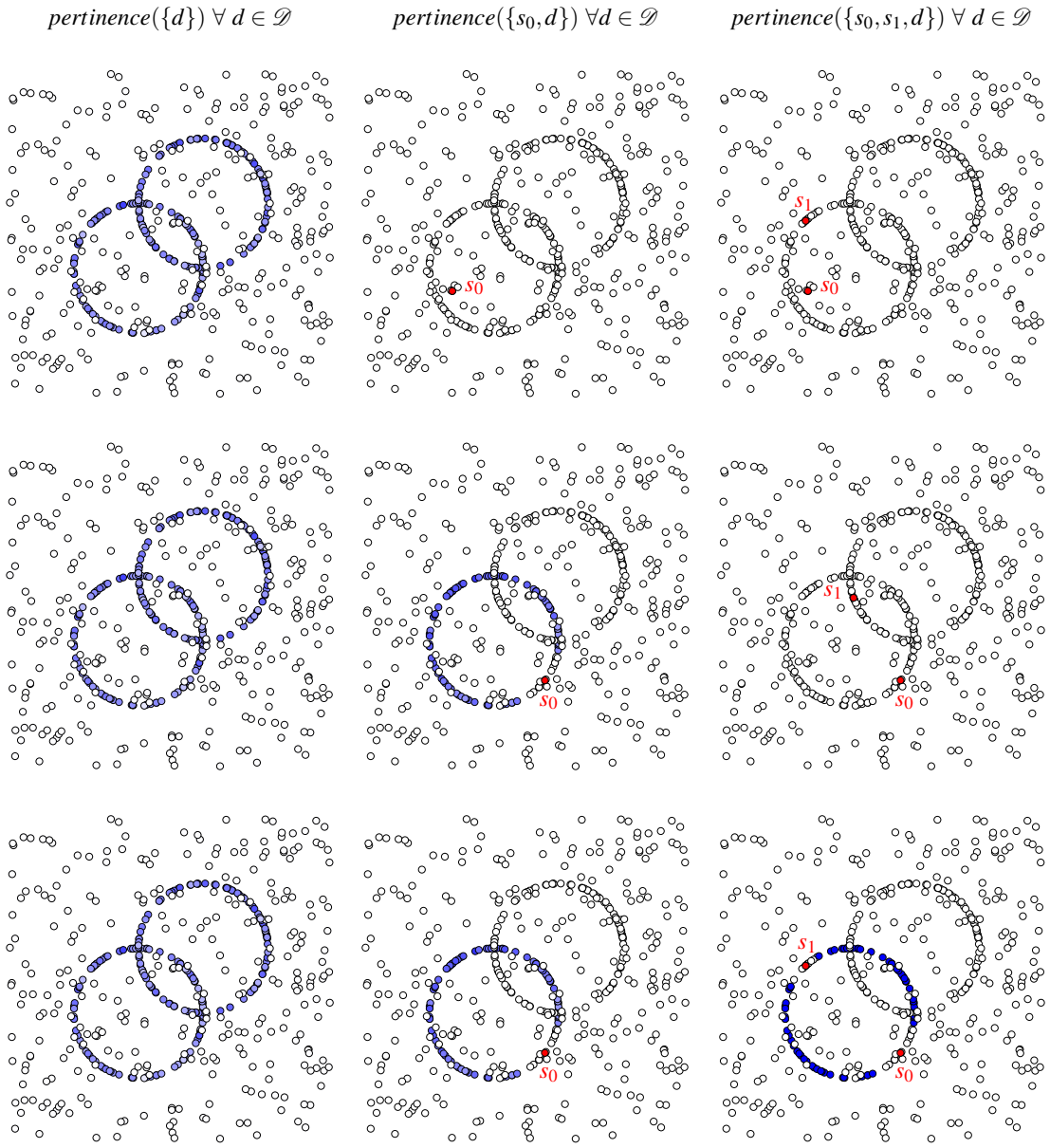


FIGURE 4.5 – Mesure de la *pertinence*, telle que définie en équation 4.10, des données dans le cas d'un problème de détection de cercles dans un nuage de points 2D. Plus une donnée est bleu, plus sa pertinence mesurée est élevée. Chaque ligne correspond à un sélection de s_0 et s_1 différente.

- $\mathbf{A}_2$) Apprentissage de l'association entre un rang dans un tri selon les valeurs de pertinence prédites par $f^l_{m\to p}$ à une pertinence réelle. Il en résulte, là encore, une fonction pour chaque taille d'échantillon $|s| = l$, appelée $\boxed{f^l_{r\to p}}$.
- $\mathbf{A}_3$) Apprentissage du vecteur $\vec{n}$ optimal, de dimension m, noté $\boxed{\vec{n}_{opti}}$.

Il est indispensable d'utiliser des ensembles de données différents pour chacun de ces apprentissages. Les fonctions $f^l_{r\to p} \forall l \in \{0,\cdots,m-1\}$ et le vecteur $\vec{n}_{opti}$ sont les seuls résultats à conserver de la partie hors ligne. La partie en ligne de OABSAC est très similaire à BetaSAC. La figure 4.8 est une description de OABSAC sous la forme d'un schéma.

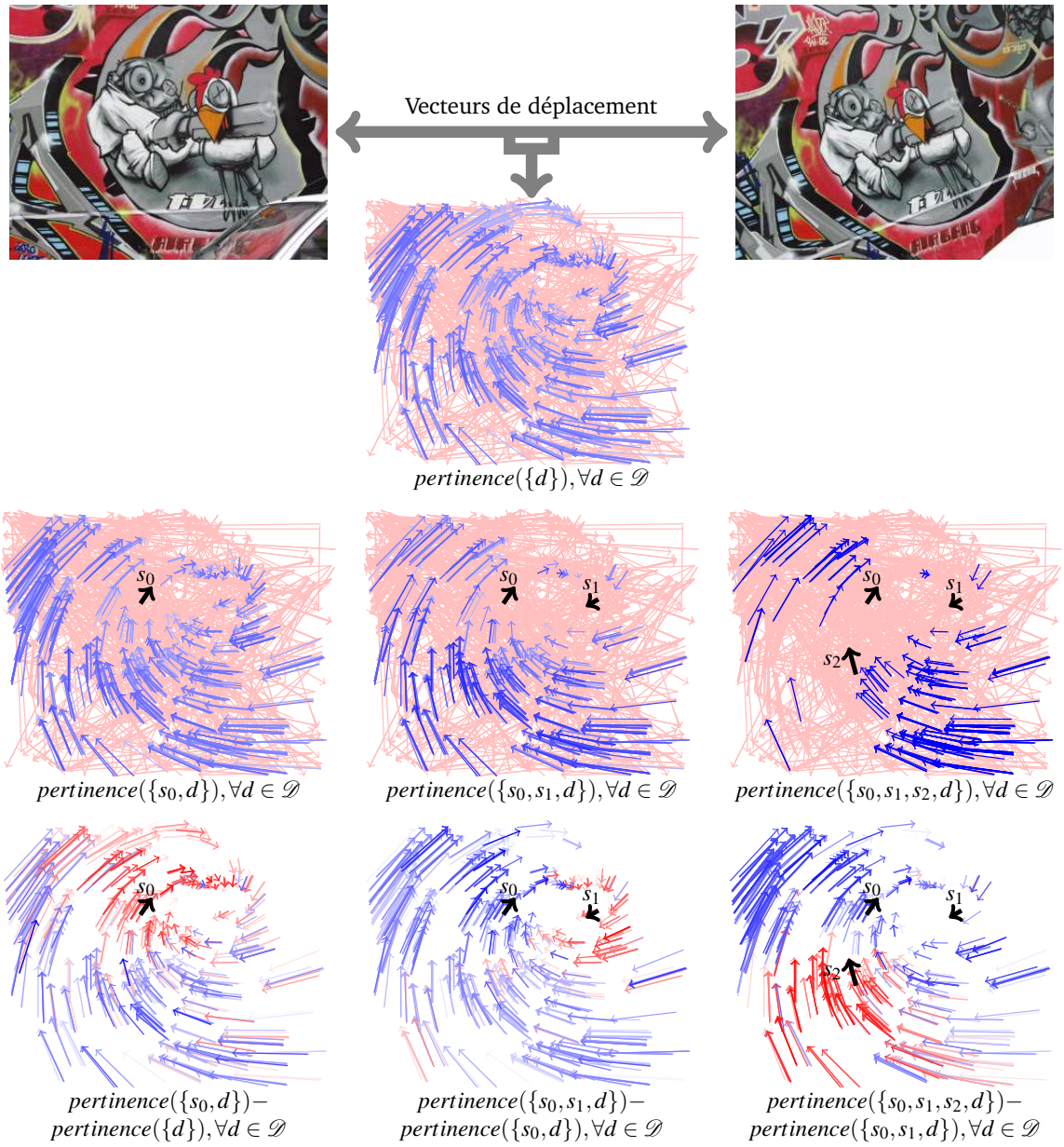


FIGURE 4.6 – Mesure de la *pertinence*, telle que définie en équation 4.10, des données dans le cas d'un problème d'estimation de l'homographie entre deux images. Ici les données visualisées sont les vecteurs de déplacement des points d'intérêt mis en correspondance. Plus une donnée est bleu, plus sa pertinence mesurée est élevée. Dans la dernière ligne, on visualise la variation des pertinences après l'ajout d'une nouvelle donnée dans l'échantillon.

4.2.3.1 Partie hors ligne

Génération des échantillons d'apprentissage La procédure de génération des échantillons d'apprentissage est la même pour chacun d'entre eux. Pour couvrir au mieux l'espace des mesures, et sachant que la pertinence de tout ensemble contaminé est nulle, on peut s'aider d'une vérité terrain et tirer plus de

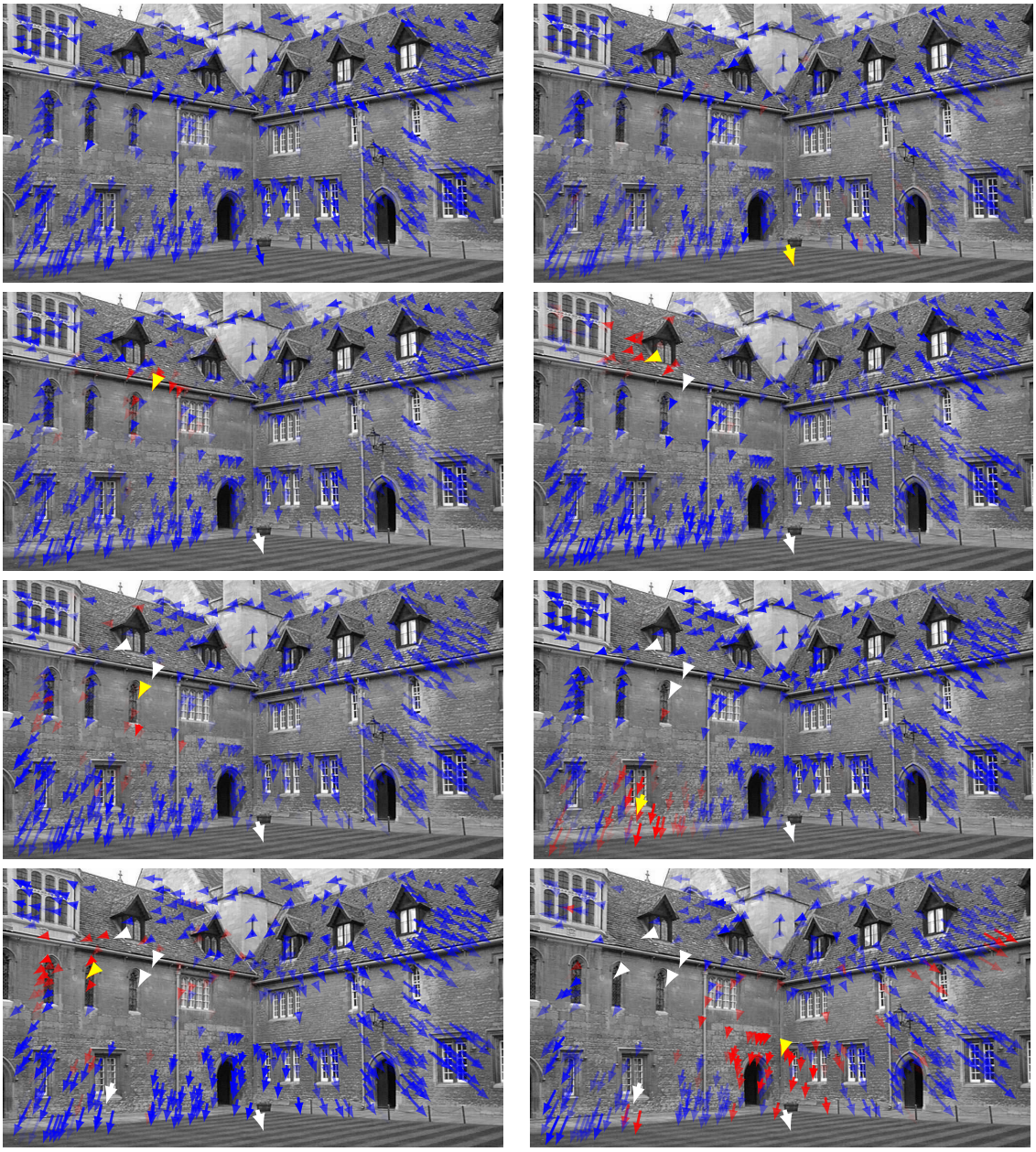


FIGURE 4.7 – Mesure de la *pertinence*, telle que définie en équation 4.10, des données dans le cas d'un problème d'estimation de la géométrie épipolaire entre deux images. Ici les données visualisées sont les vecteurs de déplacement des points d'intérêt mis en correspondance. La couleur des vecteurs correspond à leur variation de valeur de pertinence après l'ajout d'une nouvelle correspondance (en jaune) dans l'échantillon. Les vecteurs blancs sont les correspondances ajoutées précédemment dans l'échantillon.

bonnes données que ne le ferait le hasard, tout en corrigeant les valeurs d'espérance obtenues. De plus, nous cherchons à apprendre à éviter une première contamination. En effet, une fois qu'un échantillon en cours de formation est contaminé, il n'est pas intéressant d'apprendre à le compléter avec des données correctes. Ainsi nous ne générons que deux types d'échantillons, des échantillons non contaminés et des échantillons avec une unique donnée erronée. L'algorithme de génération d'échantillons de taille l est donné en 2, où U et U_1 sont respectivement les variables aléatoires uniformes dans l'ensemble des

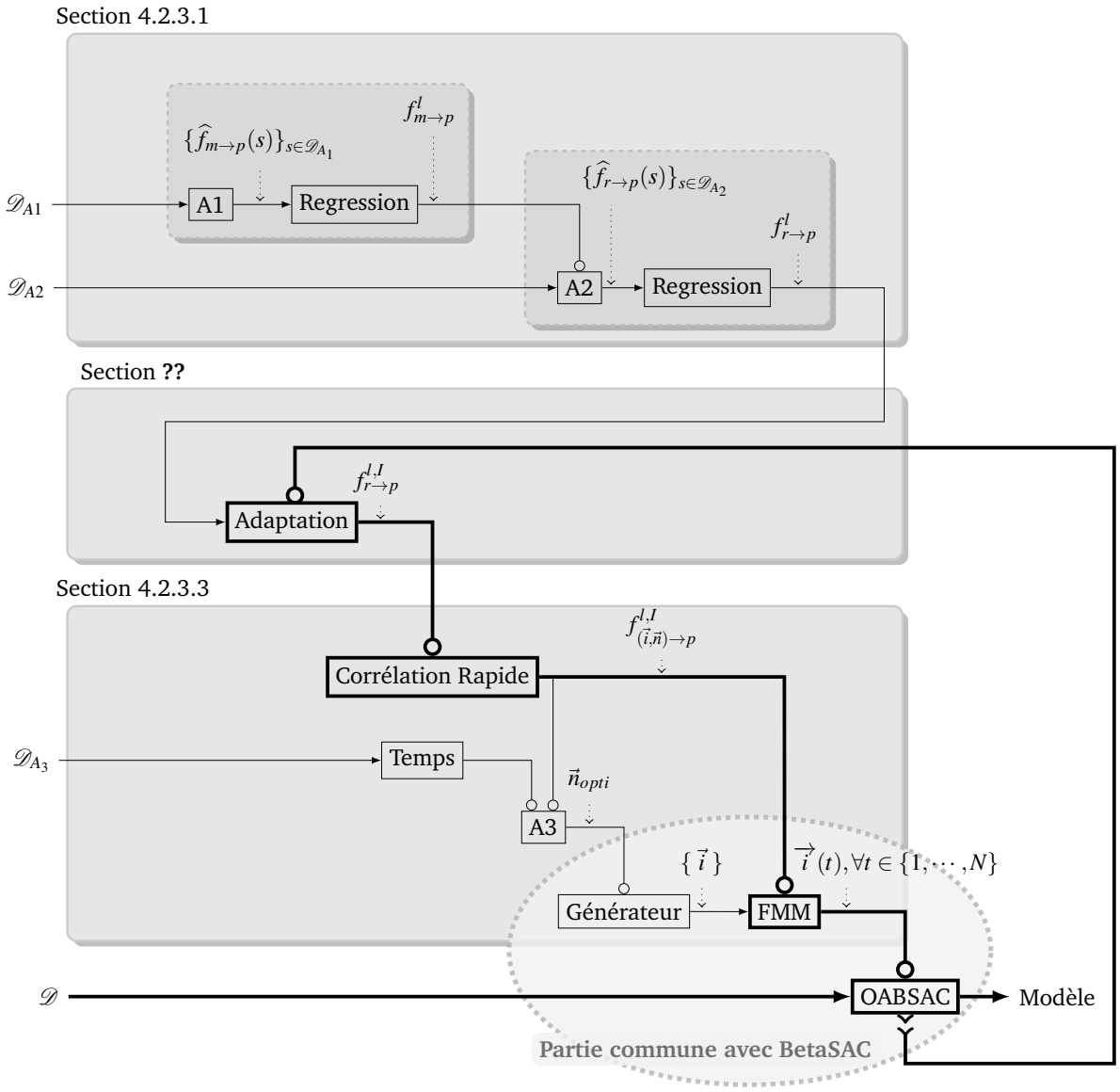


FIGURE 4.8 – Schéma général de OABSAC. La première partie décrit comment les données d'apprentissage $\mathscr{D}_{A_1}$ et $\mathscr{D}_{A_2}$ sont utilisées pour obtenir des mesures expérimentales ($\{\widehat{f}_{m\to p}(s)\}_{s\in\mathscr{D}_{A_1}}$ et $\{\widehat{f}_{r\to p}(s)\}_{s\in\mathscr{D}_{A_2}}$) généralisées en courbes résultat ($f^l_{r\to p}$ et $f^l_{r\to p}$) par régression. La seconde partie adapte éventuellement la courbe $f^l_{r\to p}$ à un taux de bonnes données I, renvoyant une courbe "déformée" $f^{l,I}_{r\to p}$. Enfin, la troisième partie présente le calcul de la stratégie optimale $\vec{n}_{opti}$ à partir des données $\mathscr{D}_{A_3}$ ainsi que la partie en ligne de OABSAC.

données, et dans l'ensemble des données correctes.

Algorithm 2: Procédure de génération d'un ensemble de N_{app} échantillons d'apprentissage de taille l.

begin
- $ensemble\_apprentissage = \{\emptyset\}$
- **for** $i = 1$ **to** N_{app} **do**
  - $s \leftarrow \{\underbrace{U_1 \cup \cdots \cup U_1}_{l-1} \cup U\}$
  - $pertinence\_s = 0$
  - **for** $j = 1$ **to** N_{esp} **do**
    - $s' \leftarrow \{\underbrace{U_1 \cup \cdots \cup U_1}_{m-l-1}\}$
    - $pertinence\_s = pertinence\_s + \frac{P(s \cup s' \text{ instancie un bon modèle})}{N_{esp}}$
- $ensemble\_apprentissage = ensemble\_apprentissage \cup \{(s, pertinence\_s)\}$

La figure 4.9 est un exemple de résultat avec $m = 4$, $N_{app} = 10$ et $N_{esp} = 3$.

Les valeurs typiquement utilisées dans nos expériences sont : $N_{app} = 3000$ et $N_{esp} = 200$, ce qui est tout à fait suffisant et rend la phase d'apprentissage relativement rapide (quelques minutes).

Apprentissage A_1 ($f^l_{m\to p}$) De même que BetaSAC, OABSAC doit être capable d'utiliser de l'information additionnelle pour parvenir à décider quelles données sont pertinentes pour compléter au mieux un échantillon partiel s. Pour n données candidates $d_1, ..., d_n$ tirées au hasard (uniforme), OABSAC estime en fait la pertinence des échantillons $s \cup \{d_1\}, ..., s \cup \{d_n\}$ afin d'ordonner les données. Pour ce faire, il associe un vecteur de *mesures* à chaque échantillon, noté $\overrightarrow{mes}(s \cup \{d\})$. L'ensemble des mesures effectuées peut être différent pour chaque taille $|s| = l$ d'échantillon. Des exemples d'informations mesurées sont donnés dans le tableau 5.3. Pour associer une valeur de pertinence à un vecteur de mesures, il nous faut apprendre cette correspondance. Cet apprentissage est en fait une simple régression effectuée dans l'espace des mesures, noté $\mathscr{M}$. En pratique nous avons utilisé la méthode RVM (Relevance Vector Machine) décrite dans l'article [52] avec, pour noyau, la fonction de base radiale gaussienne $\phi(r) = e^{-(\varepsilon r)^2}$.

$$\begin{array}{rccc} f^l_{m\to p}: & \mathscr{M} & \longrightarrow & [0,1] \\ & \overrightarrow{mes}(s) & \longmapsto & petinence(s) \end{array} \tag{4.13}$$

Fonction associant un vecteur de mesures à une valeur de pertinence.

Apprentissage A_2 ($f^s_{r\to p}$) Soit s un échantillon incomplet ($|s| < m$), et d_0 une donnée quelconque. Nous appelons rang de l'échantillon $s \cup d_0$, son rang parmi tous les échantillons $\{s \cup d\}_{d\in\mathscr{D}}$ dans le tri effectué par OABSAC, en utilisant la fonction $f^{|s|}_{m\to p}$ issue de l'apprentissage précédent. Autrement dit, si r_0 est ce rang, $d_0 = d_{(r_0)_s}$. Dans cette section, nous cherchons à apprendre la fonction $f^s_{r\to p}$, associant un rang (normalisé dans $[0,1]$) à une valeur de pertinence.

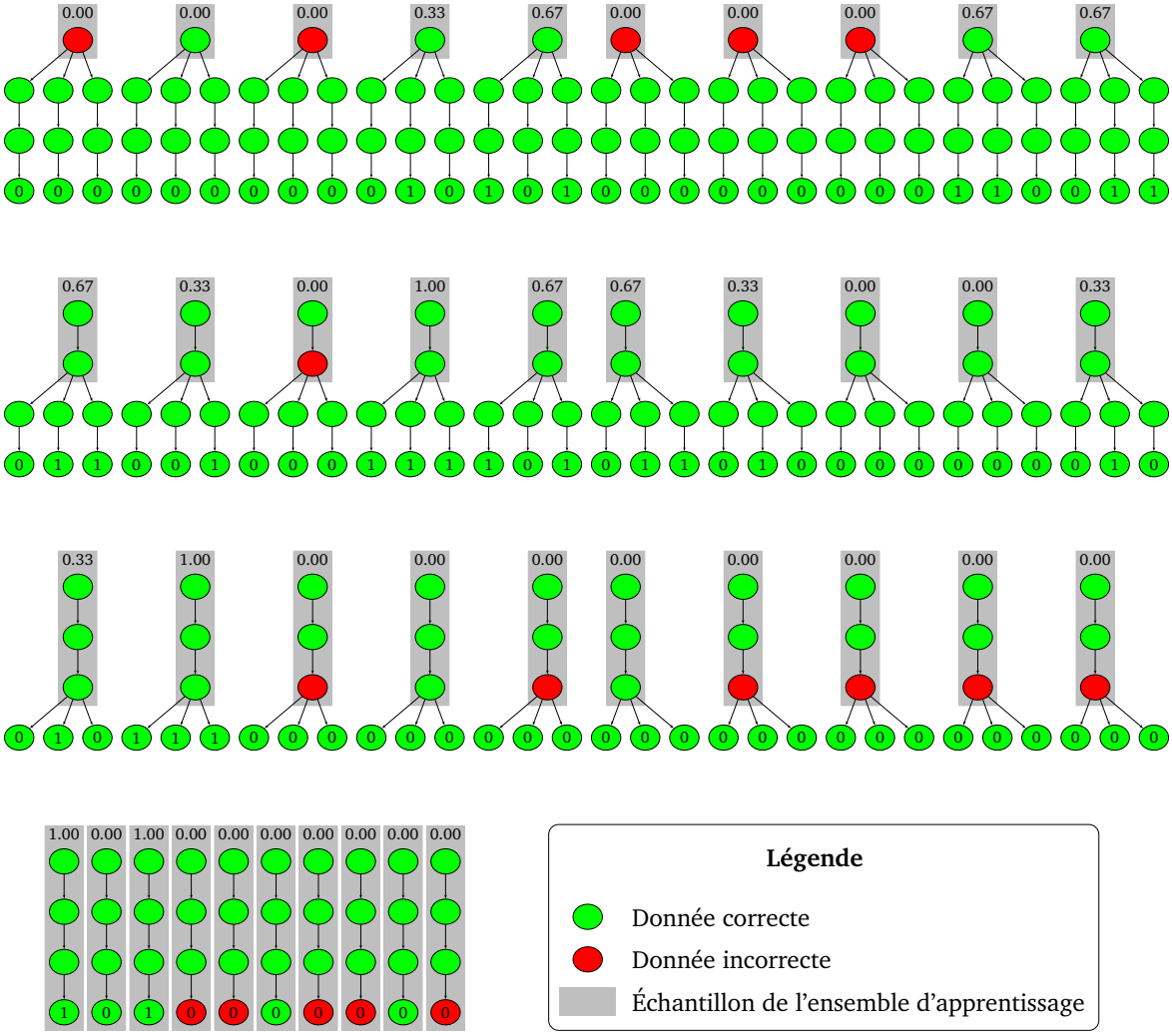


FIGURE 4.9 – Exemple d'échantillons d'apprentissage générés en utilisant la vérité terrain pour $m = 4$. Dans cet exemple, pour chaque taille d'échantillon (de 1 à 4), nous générons $N_{app} = 10$ échantillons dont on mesure la pertinence en les complétant $N_{esp} = 3$ fois. Les cercles verts et rouges représentent respectivement des bonnes et des mauvaises données, d'après la vérité terrain. Les rectangles gris représentent les échantillons d'apprentissage. Les valeurs numériques sont les pertinences mesurées des échantillons. On remarque qu'il ne suffit pas à un échantillon de ne contenir que des bonnes données pour avoir une pertinence mesurée non nulle.

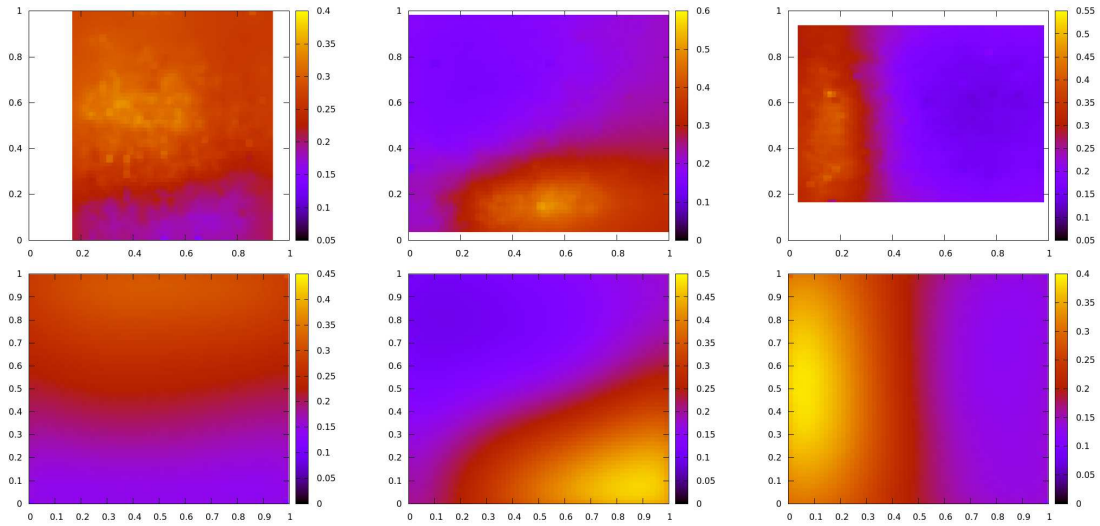


FIGURE 4.10 – Exemples de champs de valeurs de pertinence mesurés (première ligne) et généralisés par régression (seconde ligne) dans l'espace des mesures, projetés sur deux dimensions.

$$\begin{array}{rccc} f^s_{r\to p}: & [0,1] & \longrightarrow & [0,1] \\ & r & \longmapsto & pertinence\left(s \cup d_{(r)_s}\right) \end{array} \tag{4.14}$$

Fonction associant un rang à une valeur de pertinence.

Simplification La fonction $f^s_{r\to p} : r \mapsto pertinence\left(s \cup d_{(r)_s}\right)$ est définie à partir de la variable aléatoire

$$S_s(r) = s \cup d_{(r)_s} \cup \{\underbrace{U \cup \cdots \cup U}_{m-|s|-1}\},$$

où U est la variable aléatoire uniforme dans l'ensemble des données $\mathscr{D}$. En toute rigueur, la variable aléatoire $S_s(r)$ est dépendante de l'échantillon incomplet s. Cependant, nous simplifions en supposant qu'elle ne dépend que de la valeur de pertinence prédite par $f^{|s|}_{m\to p}(\overrightarrow{mes}(s))$ en distinguant deux cas :

1. si $f^{|s|}_{m\to p}(\overrightarrow{mes}(s)) = 0$, alors $f^s_{r\to p} =$ fonction nulle.
2. si $f^{|s|}_{m\to p}(\overrightarrow{mes}(s)) > 0$, alors $f^s_{r\to p} =$ fonction mesurée avec $S_s(r) = \underbrace{U_1 \cup \cdots \cup U_1}_{=s'} \cup d_{(r)_{s'}} \cup \underbrace{U_1 \cup \cdots \cup U_1}_{m-|s|-1}$ en rejetant les échantillons s dont la pertinence mesurée est nulle, avec U_1 la variable aléatoire uniforme dans l'ensemble des données correctes.

$$\begin{array}{rccc} f^{|s|}_{r\to p}: & [0,1] & \longrightarrow & [0,1] \\ & r & \longmapsto & \begin{cases} 0 & \text{si } f^{|s|}_{m\to p}(\overrightarrow{mes}(s)) = 0 \\ \mathbb{E}\left[pertinence\left(\underbrace{U_1 \cup \cdots \cup U_1}_{=s',\ |s'|=l, pertinence(s')\neq 0} \cup d_{(r)_{s'}}\right)\right] & \text{si } f^{|s|}_{m\to p}(\overrightarrow{mes}(s)) > 0 \end{cases} \end{array} \tag{4.15}$$

$$f^{s}_{r\to p} \approx f^{|s|}_{r\to p} \tag{4.16}$$

Cette simplification ne remet pas en cause la considération du caractère *dépendant* de la notion d'échantillon pertinent introduite dans notre travail. Simplement, c'est faire l'hypothèse que, après le tri en fonction des données de l'échantillon incomplet courant, la qualité d'une donnée en fonction de son emplacement dans le tri est peu dépendante du contenu de l'échantillon courant. En conséquence de cette hypothèse, nous avons l'identité 4.17.

$$\begin{aligned} P\left(S=\left\{d_{(X_0)_{\{\emptyset\}}},\cdots,d_{(X_{m-1})_{\{x_0,\cdots x_{m-2}\}}}\right\}\ pertinent\right) &= \prod_{l=0}^{m-1} P\left(d_{(X_l)_{\{x_0,\cdots x_{l-1}\}}}\ pertinent\right) \\ &= \prod_{l=0}^{m-1} \frac{f_{r\to p}(X_l)}{f_{r\to p}(U)^{\frac{m-1}{m}}} \end{aligned} \tag{4.17}$$

Conséquence de l'hypothèse **??** faite par OABSAC

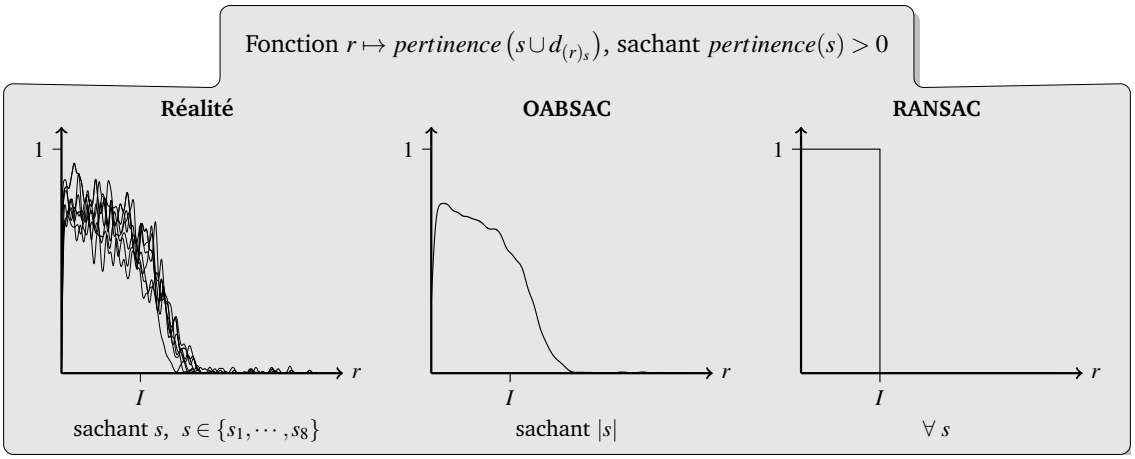


FIGURE 4.11 – La fonction de pertinence de $s \cup d$ en fonction du rang r de la donnée d est dépendante de l'échantillon s. Pour simplifier, nous supposons dans OABSAC qu'elle n'est dépendante que de deux valeurs discrètes : la taille de l'échantillon, $|s|$, et la nullité ou non de la pertinence de s.

Régression polynomiale Afin de les généraliser, les courbes $f^{l}_{r\to p}$ sont exprimées sous la forme de polynômes d'ordre p_{ordre} :

$$f^{l}_{r\to p} = \sum_{k=0}^{p_{ordre}} \lambda_{l,k}.x^{k} \tag{4.18}$$

Nous avons fait le choix des polynômes pour que la corrélation de ces fonctions avec la densité de la loi bêta soit calculable en temps constant (voir l'équation 4.19). Cette propriété est très bénéfique pour l'apprentissage A_3 et l'adaptation en ligne de OABSAC, qui font intervenir cette corrélation.

$$
\begin{aligned}
\int_0^1 f_{B_{i/n}}(x)\cdot f^l_{r\to p}(x)\,dx &= \int_0^1 f_{Beta}(x;i,n-i+1)\cdot\left[\sum_{k=0}^{p_{ordre}}\lambda_{l,k}.x^k\right]dx \\
&= \int_0^1\left[\sum_{k=0}^{p_{ordre}} f_{Beta}(x;i,n-i+1)\cdot\lambda_{l,k}.x^k\right]dx \\
&= \int_0^1\left[\sum_{k=0}^{p_{ordre}}\frac{B(i+k,n-i+1)}{B(i,n-i+1)} f_{Beta}(x;i+k,n-i+1)\cdot\lambda_{l,k}\right]dx \\
&= \int_0^1\left[\sum_{k=0}^{p_{ordre}}\frac{\frac{(i+k)!(n-i+1)!}{(k+n+1)!}}{\frac{i!(n-i+1)!}{(n+1)!}} f_{Beta}(x;i+k,n-i+1)\cdot\lambda_{l,k}\right]dx \\
&= \int_0^1\left[\sum_{k=0}^{p_{ordre}}\frac{(i+k)!(n+1)!}{i!(k+n+1)!} f_{Beta}(x;i+k,n-i+1)\cdot\lambda_{l,k}\right]dx \\
&= \int_0^1\left[\sum_{k=0}^{p_{ordre}}\left[\prod_{q=0}^{k-1}\frac{i+q}{q+n+1}\right] f_{Beta}(x;i+k,n-i+1)\cdot\lambda_{l,k}\right]dx \\
&= \left[\sum_{k=0}^{p_{ordre}}\lambda_{l,k}\left[\prod_{q=0}^{k-1}\frac{i+q}{q+n+1}\right]\right]\underbrace{\int_0^1 f_{Beta}(x;i+k,n-i+1)\,dx}_{=1} \\
&= \left[\sum_{k=0}^{p_{ordre}}\lambda_{l,k}\left[\prod_{q=0}^{k-1}\frac{i+q}{q+n+1}\right]\right]
\end{aligned}
\tag{4.19}
$$

Apprentissage A_3 ($\vec{n}_{opti}$) Contrairement à BetaSAC, dans OABSAC, n dépendant de la taille de l'échantillon en cours de formation, $l = |s|$. C'est donc un vecteur, de taille m, que nous notons $\vec{n} = [n_0, \cdots, n_{m-1}]$. Si l'on cherche à optimiser OABSAC dans le sens de la minimisation du nombre d'itérations effectuées pour résoudre un problème, alors il faut choisir $\vec{n} = [\infty, \cdots, \infty]$. Dans ce cas, chaque itération de OABSAC est coûteuse car elle nécessite un tri complet des données, mais le nombre d'itérations pour résoudre le problème sera minimal. Si l'on choisit au contraire $\vec{n} = [1, \cdots, 1]$, OABSAC se comportera exactement comme RANSAC. Si l'on souhaite optimiser le temps (en secondes) de résolution d'un problème, alors le vecteur optimal, noté $\vec{n}_{opti}$ doit se situer entre les deux. Ce vecteur dépend de deux choses :

- Le temps consommé par chaque étape, pour un matériel et une implémentation donnée de l'algorithme.
- Le résultat de l'apprentissage précédent : $\left\{f^l_{r\to p}\right\}_{l\in\{0,\cdots,m-1\}}$

Trouver ce vecteur optimal $\vec{n}_{opti}$ revient à résoudre une optimisation discrète dont la fonction est convexe (pas théoriquement mais quasiment toujours en pratique). Sa résolution est donc facile par une méthode de type *descente de gradient* avec des pas discrets. En revanche, elle est relativement longue puisque chaque calcul du temps théorique total pour un vecteur $\vec{n}$ a pour complexité algorithme $O(T\cdot m\cdot p^2_{ordre})$, où p_{ordre} est l'ordre des polynôme utilisés pour représenter les fonctions $f_{r\to p}$ (voir l'équation 4.19). Pour réaliser ce calcul, nous devons considérer les temps consommés par chaque étape de l'algorithme. Nous utilisons la lettre c (comme *chronos*) pour représenter les temps en secondes sans confondre avec le compteur d'itération t. La liste des temps pris en compte est présentée dans le tableau 4.1.

Les calculs suivants permettent d'obtenir l'espérance de la durée totale de la résolution du problème 4.20. Dans notre implémentation, nous utilisons, pour les calculs intermédiaires, les formulations par récurrences (équations 4.23 et 4.25) plutôt que les formulations directes (équations 4.22 et 4.24), ce qui les rend relativement rapides. Comme dans les parties précédentes, les lettres capitales désignent des variables aléatoires.

Durée totale Soit C_{fin} la variable aléatoire de la durée totale (en millisecondes) de l'estimation des paramètres du modèle recherché. Nous calculons son espérance en équation 4.20. C'est cette expression que nous cherchons à minimiser par le choix du vecteur $\vec{n}$.

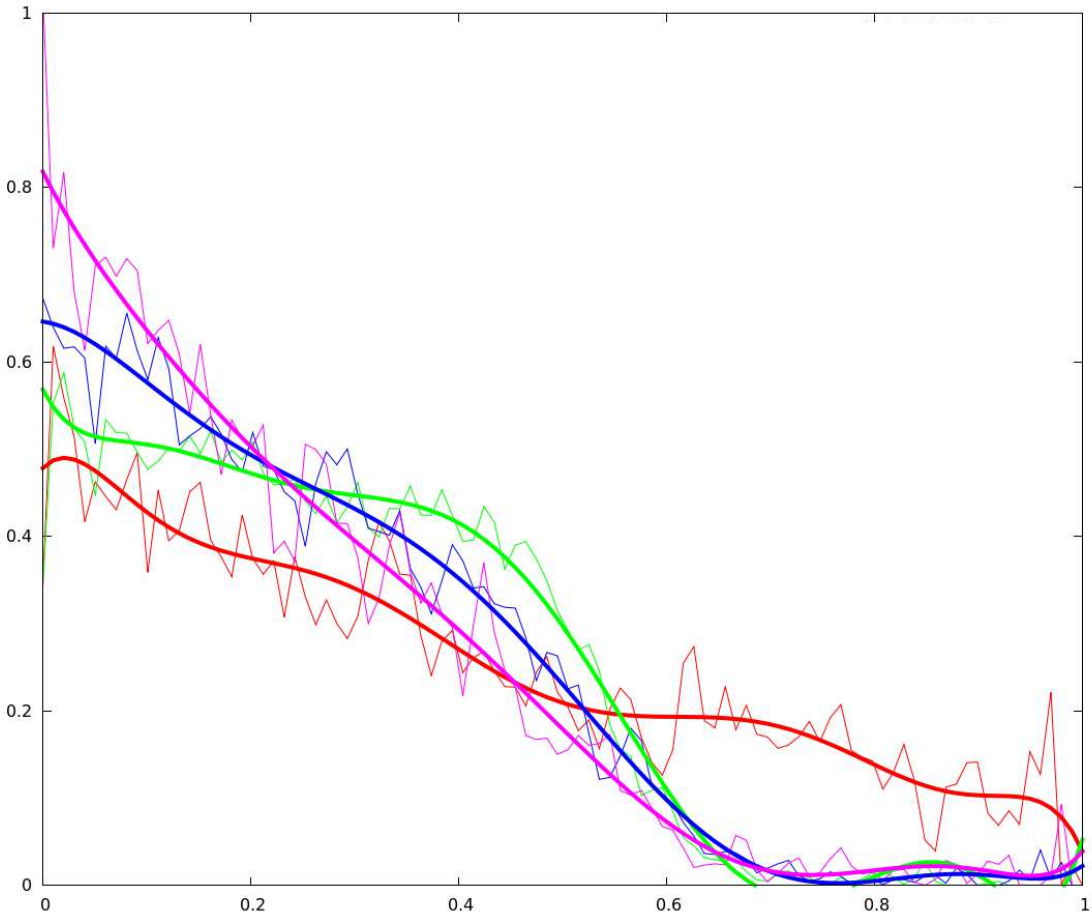


FIGURE 4.12 – Exemple d'ensemble de fonctions $\left\{f^{l=0}_{r\to p}(rouge), f^{l=1}_{r\to p}(vert), f^{l=2}_{r\to p}(bleu), f^{l=3}_{r\to p}(violet)\right\}$ mesurées et interpolées par des polynômes de degré 12.

Notation	Signification
c_s	Sélection d'un échantillon de taille m
$c_{mes}(l)$	Calcul de l'ensemble des mesures correspondant à $\|m\| = l$
$c_{k^{th}}(n)$	Recherche du k-ième élément parmi n
c_t	Test du modèle-hypothèse sur une donnée
c_{fin}	Déroulement complet jusqu'à ce qu'une solution au problème soit trouvée

TABLE 4.1 – Liste des temps mesurés pendant la phase d'apprentissage

$$\begin{aligned}
\mathbb{E}\left[C_{fin}\right] &= \left(\sum_{c=0}^{\infty} c \cdot P(C_{fin} = c)\right) \\
&= \left(\sum_{c=0}^{\infty} c \cdot \left(\sum_{t=1}^{\infty} P(T_{fin} = t) \cdot P(C_{fin} = c \mid T_{fin} = t)\right)\right) \\
&= \left(\sum_{t=0}^{\infty} P(T_{fin} = t) \cdot \left(\sum_{c=0}^{\infty} c \cdot P(C_{fin} = c \mid T_{fin} = t)\right)\right) \\
&= \left(\sum_{t=0}^{\infty} P(T_{fin} = t) \cdot \mathbb{E}\left[C_{fin} \mid T_{fin} = t\right]\right)
\end{aligned} \tag{4.20}$$

Cette somme converge asymptotiquement vers l'espérance recherchée lorsque t tend vers $+\infty$. En pratique, dans notre cas, nous pouvons interrompre son évaluation à $t = 3 \cdot N$.

$$\mathbb{E}\left[C_{fin}\right] \approx \left(\sum_{t=0}^{3 \cdot N} P(T_{fin} = t) \cdot \mathbb{E}\left[C_{fin} \mid T_{fin} = t\right]\right) \tag{4.21}$$

Durée totale connaissant l'itération finale

Formulation directeFormulation directe

$$\begin{aligned}
\mathbb{E}\left[C_{fin} \mid T_{fin} = t_{fin}\right] &= \mathbb{E}\left[\sum_{t=1}^{t_{fin}} C(t) \mid T_{fin} = t_{fin}\right] \\
&= \sum_{t=1}^{t_{fin}} \mathbb{E}\left[C(t) \mid T_{fin} = t_{fin}\right] \\
&= \left(\sum_{t=1}^{t_{fin}-1} \mathbb{E}\left[C(t) \mid T_{fin} \neq t\right]\right) + \mathbb{E}\left[C(t_{fin}) \mid T_{fin} = t_{fin}\right] \\
&= \left(\sum_{t=1}^{t_{fin}-1} P(S(t) \text{ pertinent}) \cdot \mathbb{E}\left[C(t) \mid T_{fin} \neq t \ \&\ S(t) \text{ pertinent}\right]\right. \\
&\qquad \left. + (1 - P(S(t) \text{ pertinent})) \cdot \mathbb{E}\left[C(t) \mid T_{fin} \neq t \ \&\ S(t) \text{ non pertinent}\right]\right) \\
&\qquad + \mathbb{E}\left[C(t_{fin}) \mid T_{fin} = t_{fin}\right]
\end{aligned} \tag{4.22}$$

Formulation par récurrenceFormulation par récurrence

$$\begin{aligned}
\mathbb{E}\left[C_{fin} \mid T_{fin} = t_{fin} + 1\right] &= \left(\sum_{t=1}^{t_{fin}} \mathbb{E}\left[C(t) \mid T_{fin} \neq t\right]\right) + \mathbb{E}\left[T(t_{fin}+1) \mid T_{fin} = t_{fin} + 1\right] \\
&= \mathbb{E}\left[C_{fin} \mid T_{fin} = t_{fin}\right] - \mathbb{E}\left[C(t_{fin}) \mid T_{fin} = t_{fin}\right] + \mathbb{E}\left[C(t_{fin}+1) \mid T_{fin} = t_{fin} + 1\right] \\
&\quad + P(S(t) \text{ pertinent}) \cdot \mathbb{E}\left[C(t_{fin}) \mid T_{fin} \neq t_{fin} \ \&\ S(t) \text{ pertinent}\right] \\
&\quad + (1 - P(C(t) \text{ pertinent}) \cdot \mathbb{E}\left[C(t_{fin}) \mid T_{fin} \neq t_{fin} \ \&\ S(t) \neg\text{pertinent}\right]
\end{aligned} \tag{4.23}$$

Itération finale

Formulation directeFormulation directe

$$\begin{aligned}
P(T_{fin} = t_{fin}) &= \left(\prod_{t=1}^{t_{fin}-1} P(\acute{E}CHEC(t))\right) . P(SUCC\grave{E}S(t_{fin})) \\
&= \left(\prod_{t=1}^{t_{fin}-1} P\left(S(t) \text{ non pertinent} + S(t) \text{ pertinent } \&\ TEST(t) = 0\right)\right) . P(S(t_{fin}) \text{ pertinent } \&\ TEST(t_{fin}) = 1) \\
&= \left(\prod_{t=1}^{t_{fin}-1} 1 - P(S(t) \text{ pertinent}) + P(S(t) \text{ pertinent } \&\ TEST(t) = 0)\right) . P(S(t_{fin}) \text{ pertinent } \&\ TEST(t_{fin}) = 1) \\
&= \left(\prod_{t=1}^{t_{fin}-1} 1 + P(S(t) \text{ pertinent}) . (P(TEST_{FN}(t)) - 1)\right) . P(S(t_{fin}) \text{ pertinent}) . P(TEST_{TP}(t_{fin}))
\end{aligned} \tag{4.24}$$

Formulation par récurrenceFormulation par récurrence

$$\begin{aligned}
P(T_{fin} = t_{fin} + 1) &= P(T_{fin} = t_{fin}) \cdot \frac{P(S(t_{fin}+1) \text{ pertinent}) \cdot P(TEST_{TP}(t_{fin}+1))}{P(S(t_{fin}) \text{ pertinent}) \cdot P(TEST_{TP}(t_{fin}))} \\
&\quad \cdot \left(1 + P(S(t_{fin}) \text{ pertinent}) \cdot (P(TEST_{FN}(t_{fin})) - 1)\right)
\end{aligned} \tag{4.25}$$

Formation d'un échantillon pertinent

$$P(S(t) \text{ pertinent}) = \prod_{l=0}^{m-1} P\left(X_{i(t,l)/n(l)} \text{ pertinent}\right) \tag{4.26}$$

Durée d'une itération

$$\mathbb{E}[C(t) \mid X] = c_s + \left(\sum_{l=0}^{m-1} n(l) \cdot c_{mes}(l) + c_{k^{th}}(n)\right) + c_t \cdot \mathbb{E}[NB_t(t) \mid X], \quad \forall X \tag{4.27}$$

4.2.3.2 Adaptation au taux d'erreurs

Nous supposons que l'allure des fonctions $\left\{f^l_{r\to p}\right\}_{l\in\{0,\cdots,m-1\}}$ est intrinsèque au problème et aux mesures utilisées, alors que le taux de données correctes, I, ne l'est pas. Alors, le but de cette section est de rendre la partie hors ligne de OABSAC indépendante de I, et la partie en ligne adaptable au taux I. Pour cela, nous cherchons la transformation $T_{I_1\to I_2}$ capable d'adapter une fonction $f_{r\to p}$ quelconque à un taux différent de celui rencontré pendent la partie hors ligne.

Nous notons $f^I_{r\to p}$, la fonction adaptée à un taux I, donc, $f^{I_2}_{r\to p} = T_{I_1\to I_2}(f^{I_1}_{r\to p})$. $T_{I_1\to I_2}$ doit vérifier les propriétés 4.28, où 1_F est la fonction indicatrice.

$$\forall\ \beta,\delta,I,I_1,I_2 \in [0,1], \qquad \begin{array}{ll} \textbf{1.} & T_{I\to I} = \mathbb{1} \\ \textbf{2.} & T_{I_1\to I_2} \circ T_{I_2\to I_1} = \mathbb{1} \\ \textbf{3.} & T_{I_1\to I_2}(1_{[0,\delta]}) = 1_{[0,\frac{I_2}{I_1}\delta]} \\ \textbf{4.} & T_{I_1\to I_2}(k\mapsto\beta) = k\mapsto\dfrac{I_2}{I_1}\beta \\ \textbf{5.} & \displaystyle\int_{k=0}^{1} T_{I_1\to I_2}(f^{I_1}_{r\to p})(k)\,dk \;=\; \frac{I_2}{I_1}\int_{k=0}^{1} f^{I_1}_{r\to p}(k)\,dk \end{array} \tag{4.28}$$

Pour déterminer la transformation $T_{I_1\to I_2}$ (équation 4.29), nous avons raisonné sur l'exemple discret donné en tableau 4.2.3.2, avec $I_1 = 0.5$ et $I_2 = 0.75$. C'est-à-dire que nous avons imaginé une fonction $f^{I_1=0.5}_{r\to p}(r_1)$ mesurée à partir de 6 réalisations. Chacune des réalisations a une valeur de pertinence discrète (donc soit 0, soit 1). Alors, dans ce cas discret, la transformation $T_{I_1\to I_2}$ qui remplace tous les 1 par une succession de plusieurs 1 (dont le nombre vaut $\gamma_{I_1,I_2} = \frac{1-I_1}{I_1}.\frac{I_2}{1-I_2}$) vérifie bien les propriétés souhaitées (équation 4.28). La transformation que nous utilisons est simplement son extrapolation continue. Quelques exemples graphiques du comportement de cette transformation sont donnés en figure 4.13.

r_1	0			1			2	3			4	5
$f^{I_1=0.5}_{r\to p}(r_1)$	1			1			0	1			0	0
$f^{I_2=0.75}_{r\to p}(r_2)$	1	1	1	1	1	1	0	1	1	1	0	0
r_2	0	1	2	3	4	5	6	7	8	9	10	11

TABLE 4.2 – Exemple discret utilisé pour mettre au point notre transformation $T_{I_1\to I_2}$, permettant de synthétiser une fonction $f^{I_2}_{r\to p}(r)$ à partir d'une fonction observée $f^{I_1}_{r\to p}(r)$. On passe d'un taux de positifs I_1 à I_2 en multipliant les positifs par le facteur γ_{I_1,I_2} de l'équation 4.29

.

$$f^{I_2}_{r\to p}(r_2) = \frac{\gamma_{I_1,I_2} \cdot p_{I_1}\left(r1r2^{-1}(r_2)\right)}{1+(\gamma_{I_1,I_2}-1)\cdot f^{I_1}_{r\to p}\left(r1r2^{-1}(r_2)\right)}$$

avec :

$$\gamma_{I_1,I_2} = \frac{1-I_1}{I_1}\cdot\frac{I_2}{1-I_2} \qquad \text{et} \qquad r1r2(r_1) = r_1 + (\gamma_{I_1,I_2}-1)\int_0^{r_1} f^{I_1}_{r\to p}(r)\,dr \tag{4.29}$$

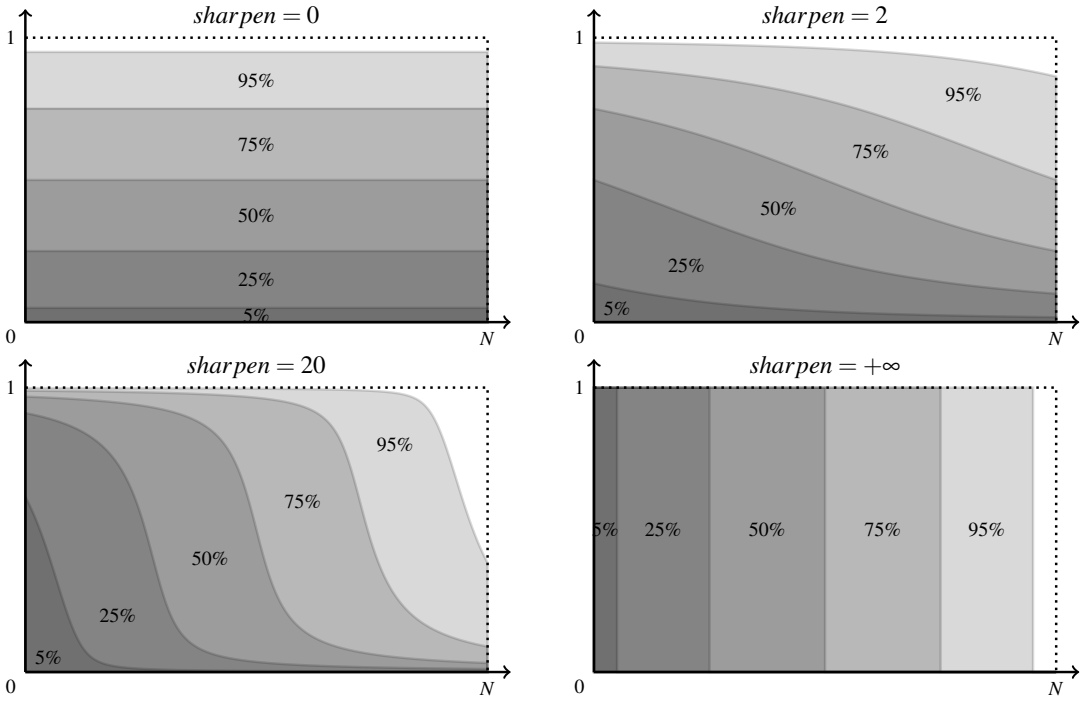


FIGURE 4.13 – Exemples d'ensembles de fonctions $\left\{f^I_{r\to p}\right\}_{I\in\{5\%,25\%,50\%,75\%,95\%\}}$, calculés à partir de la fonction $f^{50\%}_{r\to p}(r) = 1-(\frac{1}{\pi}tan^{-1}\left(sharpen*(r-\frac{1}{2})\right)+\frac{1}{2})$ pour quatre valeurs de *sharpen* différentes : $0, 2, 20$ et $+\infty$.

4.2.3.3 Partie en ligne

Ordonnancement des vecteurs de stratégie Cette étape est inchangée par rapport à BetaSAC si ce n'est que les vecteurs ne sont plus ordonnés selon leur p-norme mais selon leur espérance de pertinence, déduite de l'ensemble des fonctions $\left\{f^l_{r\to p}\right\}_{l\in\{0,\cdots,m-1\}}$ en utilisant l'équation 4.30. L'arbre parcouru par la méthode de *fast marching* est également le même (figure 4.3).

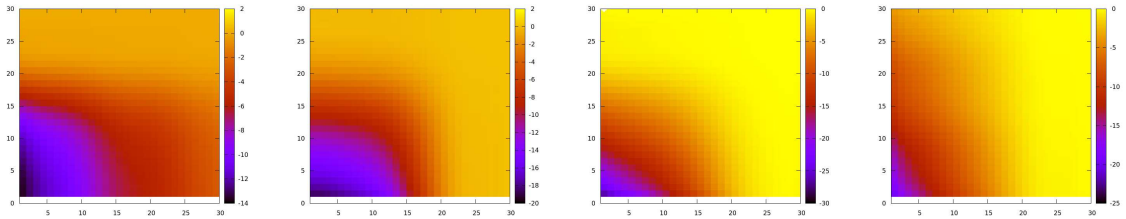


FIGURE 4.14 – Exemple de résultat : valeurs $P\left(S_{[i_0,\ldots,i_{m-1}]} \text{ pertinent}\right)$ de la probabilité de tirer un échantillon pertinent en fonction de $i_0,\ldots,i_{m-1}$, projetées sur différents couples de dimensions. On remarque la proximité avec la norme 3, utilisée dans BetaSAC.

$$\begin{aligned} P\left(S_{[i_0,\ldots,i_{m-1}]} \text{ pertinent}\right) &= \prod_{l=0}^{m-1}\left[\int_0^1 f_{P_{B_{i/n}}}(x).f_{P\left(S_{[i_0,\cdots,i_{l-1}]}\cup d_{(x)s}\cup S_{[i_{l+1},\cdots,i_{l+1}]} \text{ pertinent}\right)}dx\right] \\ &\approx\propto \prod_{l=0}^{m-1}\left[\int_0^1 f_{P_{B_{i/n}}}(x).f_{P\left(S_U\cup d_{(x)s}\cup S_U \text{ pertinent}\right)}dx\right] \\ &\approx\propto \prod_{l=0}^{m-1}\left[\sum_{k=0}^{p_{ordre}}\lambda_{l,k}\left(\prod_{q=0}^{k-1}\frac{i_l+q}{n_l+q+1}\right)\right] \end{aligned} \tag{4.30}$$

Valeur proportionnelle à la probabilité de pertinence d'un échantillon formé en utilisant le vecteur de sélection $[i_0,\ldots,i_{m-1}]$ en utilisant l'hypothèse 4.16

Adaptation en ligne On peut réordonnancer en ligne les vecteurs de stratégie à venir sans violer les propriétés $\mathscr{P}_1$ et $\mathscr{P}_2$ de l'équation 3.5. On peut donc intégrer toute information acquise pendant les itérations de OABSAC, pour optimiser en ligne sa stratégie. En pratique, il n'y a pas de tri à effectué, il s'agit juste de faire dépendre la fonction de calcul de priorité de la FMM, de variables dont la valeur change au cours des itérations. Le coût supplémentaire est donc nul.

La première information à laquelle on pense est le taux de données correctes I, dont on peut facilement estimer une borne inférieure en fonction du score maximum obtenu jusque là, comme expliqué dans la section 2.2.3.

$$I \geq I_{observé}(t) = \frac{ScoreMax(t)}{N} \tag{4.31}$$

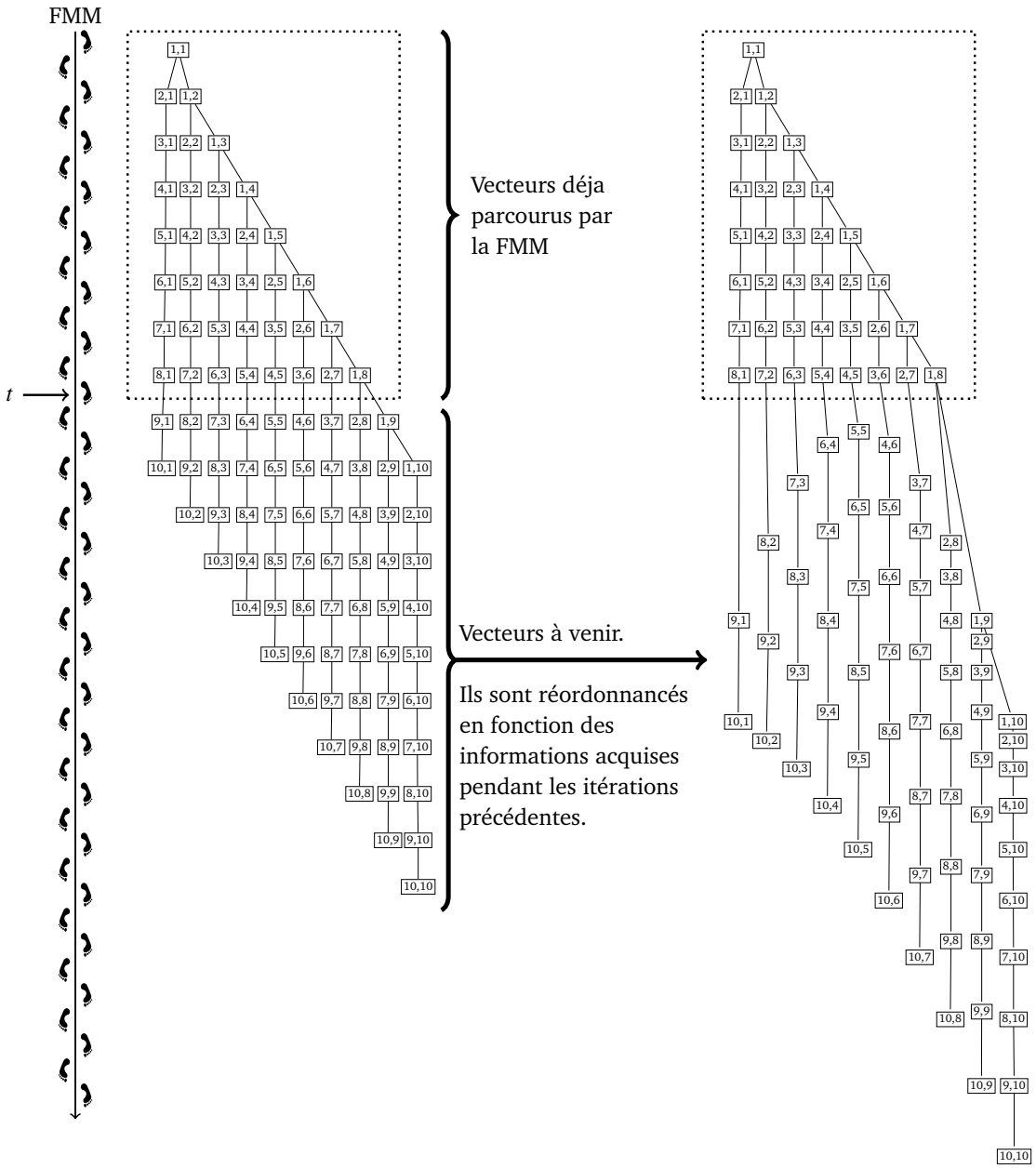


FIGURE 4.15 – Réordonnancement en ligne des vecteurs de sélection à venir. Dans cet exemple, $m = 2$ et $\vec{n} = (10, 10)$. Voir la figure 4.3 pour plus de détails.

5

Applications et résultats

Sommaire

Dans cette section, nous rendons compte des expérimentations conduites dans le cadre de l'estimation d'homographies et de matrices fondamentales à partir d'un couple de photographies d'une même scène. Au cours des phases hors ligne et en ligne, nous considérons que le problème est résolu lorsque l'on a trouvé un modèle expliquant un taux $\alpha = 75\%$ des données considérées comme correctes d'après la vérité terrain que nous possédons.

5.1 Mesure de la stratégie

L'approche proposée dans ce travail est de sélectionner les meilleures données en premier, de façon à former les meilleurs échantillons. L'ordonnancement des données est effectué à partir de l'ensemble des mesures, capables de nous fournir une information utile sur elles. L'ordonnancement des échantillons est, quant à lui, assuré par la méthode Fast Marching. Ce sont donc deux étapes bien distinctes qu'il est nécessaire d'évaluer séparément.

5.1.1 Ordonnancement des données

Pour chaque mesure, nous présentons la courbe de la probabilité de tirer une donnée pertinente en fonction de son rang dans le tri effectué sachant le contenu actuel de l'échantillon s :

$$f_{r\to p} : \quad \mathbf{r} \mapsto \mathit{pertinence}\left(s \cup \{d_{(\mathbf{r})_s}\}\right) \tag{5.1}$$

Ces courbes sont obtenues en mesurant la pertinence (telle que définie par 4.10) moyenne sur 5000 échantillons. Plus de détails sur la façon d'obtenir ces courbes sont donnés dans le paragraphe 4.2.3.1.

Homographie Les résultats dans le cas de la recherche des paramètres d'une homographie sont présentés dans le tableau 5.1. La base d'images d'apprentissage utilisée est un ensemble de couples de photographies d'objets plans, prises de points de vue différents. Les images utilisées présentent une homographie plus ou moins marquée. Nous possédons la vérité terrain de chaque couple d'images, ce qui nous permet

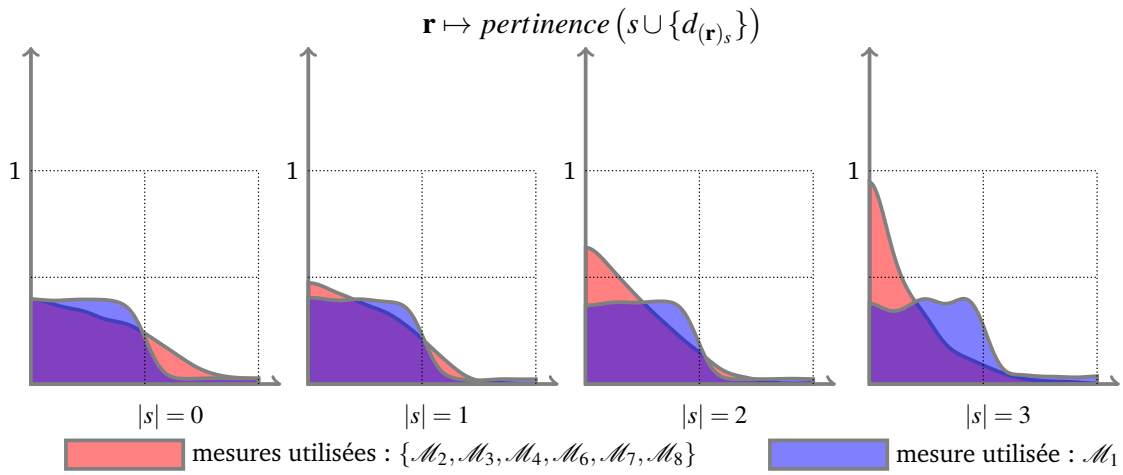


FIGURE 5.1 – Courbes de la probabilité de tirer une donnée pertinente en fonction de son rang dans le tri (équation 5.1), pour deux tris différents. En rouge : tri effectué à partir d'un ensemble de mesures utilisées par OABSAC (décrites dans le tableau 5.3). En bleu : tri effectué à partir de la vérité terrain (de type donnée correcte/donnée incorrecte). On constate que les mesures effectuées par OABSAC sont beaucoup plus informatives qu'une vérité terrain. Ces mesures ont été effectuées dans le cadre de l'estimation d'une homographie, sur un ensemble de quelques images représentatives. On obtient des résultats très similaires pour l'estimation de la matrice fondamentale.

de forcer un taux de correspondances correctes I à 50% pour chaque couple. La figure 5.1, compare les résultats obtenus avec l'information supplémentaire que nous proposons d'utiliser et ceux que l'on obtient en utilisant simplement une vérité terrain, dans laquelle chaque donnée est bonne ou mauvaise. On constate que l'information utilisée est nettement meilleure qu'une vérité terrain de ce type.

Géométrie épipolaire Les résultats dans le cas de la recherche des paramètres d'une géométrie épipolaire sont présentés dans le tableau 5.2. La base d'images d'apprentissage utilisée est un ensemble de couples de photographies de scènes d'extérieur, prises de points de vue différents. Les écarts de points de vue sont plus ou moins importants. Nous possédons la vérité terrain de chaque couple d'images, ce qui nous permet de forcer un taux de correspondances correctes I à 50% pour chaque couple.

5.1.2 Ordonnancement des vecteurs de sélection

Dans la section précédente, nous avons validé le bénéfice de notre ordonnancement des données à partir d'un ensemble de mesures. Dans cette section, nous vérifions que notre stratégie de sélection est capable de générer les meilleurs échantillons complets (de taille $m = 4$ pour ces mesures) dans les premières itérations.

La figure 5.2 présente la probabilité de former un échantillon complet pertinent au cours des itérations de BetaSAC. Cette probabilité est calculée à partir de trois qualités de tris théoriques $f_{r\to p_1}, f_{r\to p_2}$ et $f_{r\to p_3}$. Pour chacune de ces fonctions, la probabilité de former un échantillon complet pertinent est calculée en supposant que la qualité du tri ne change pas au cours de la création de l'échantillon. Autrement dit, $f^{l=0}_{r\to p_1} = f^{l=1}_{r\to p_1} = f^{l=2}_{r\to p_1} = f^{l=3}_{r\to p_1} = f_{r\to p_1}$, et de même pour $f_{r\to p_2}$ et $f_{r\to p_3}$.

La figure 5.3 illustre le bénéfice de OABSAC par rapport au tri effectué par BetaSAC. Cette figure est calculée en supposant cette fois que la qualité du tri est très différente selon la taille, l, de l'échantillon en cours de formation. On constate que dans ce cas, la stratégie de OABSAC est optimale alors que celle de BetaSAC est loin de l'être.

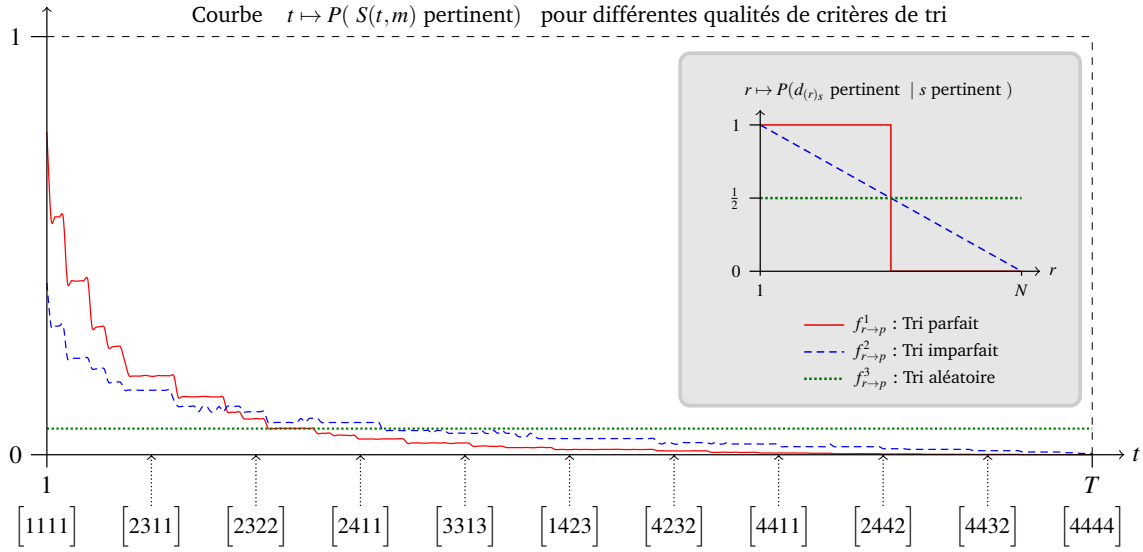


FIGURE 5.2 – Probabilité de générer un bon échantillon pendant les itérations de BetaSAC en supposant des tris plus ou moins parfaits décrits dans l'encart (en haut à droite). Les paramètres utilisés sont $n = 4$ et $p = 3$. La taille d'un échantillon (m) vaut 4 et le taux de données correctes est 50%. Le vecteur de sélection $[i_0(t), \ldots, i_{m-1}(t)]$ est indiqué pour quelques valeurs d'itération t.

5.2 Comparaison avec les méthodes précédentes

A présent, nous comparons les performances de BetaSAC et OABSAC pour l'estimation d'une homographie et d'une géométrie épipolaire avec celles de RANSAC et PROSAC.

5.2.1 Estimation d'une homographie

Le tableau 5.4 est un compte rendu d'une série d'expériences d'estimation d'homographies par les méthodes RANSAC, PROSAC, BetaSAC et OABSAC.

5.2.2 Estimation d'une géométrie épipolaire

Nous présentons dans les tableaux 5.5, 5.6 et 5.7 les résultats des tests réalisés sur la base d'images issue de [63]. Nous n'évaluons par BetaSAC sur le problème de l'estimation d'une géométrie épipolaire, car nous n'avons pas proposé de critère de tri des données pour ce cas spécifique.

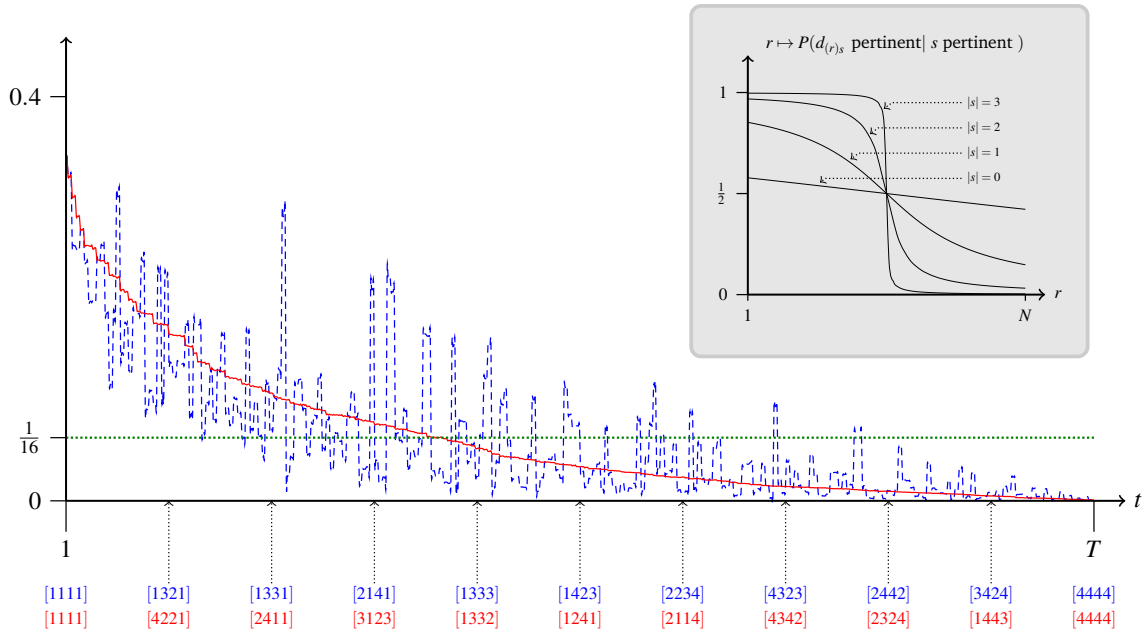


FIGURE 5.3 – Probabilité de générer un bon échantillon pendant les itérations de RANSAC (en vert), BetaSAC (en bleu) et OABSAC (en rouge) en supposant des tris plus ou moins parfaits, décrits dans l'encart (en haut à droite). Les paramètres utilisés sont $n = 4$ et $p = 3$ pour BetaSAC et $\vec{n} = [4,4,4,4]$ pour OABSAC. La taille d'un échantillon (m) vaut 4 et le taux de données correctes est 50%. Le vecteur de sélection $[i_0(t), \ldots, i_{m-1}(t)]$ est indiqué pour quelques valeurs d'itération t. Il est clair que l'ordonnancement des vecteurs de sélection effectué par OABSAC est bien meilleur que celui de BetaSAC.

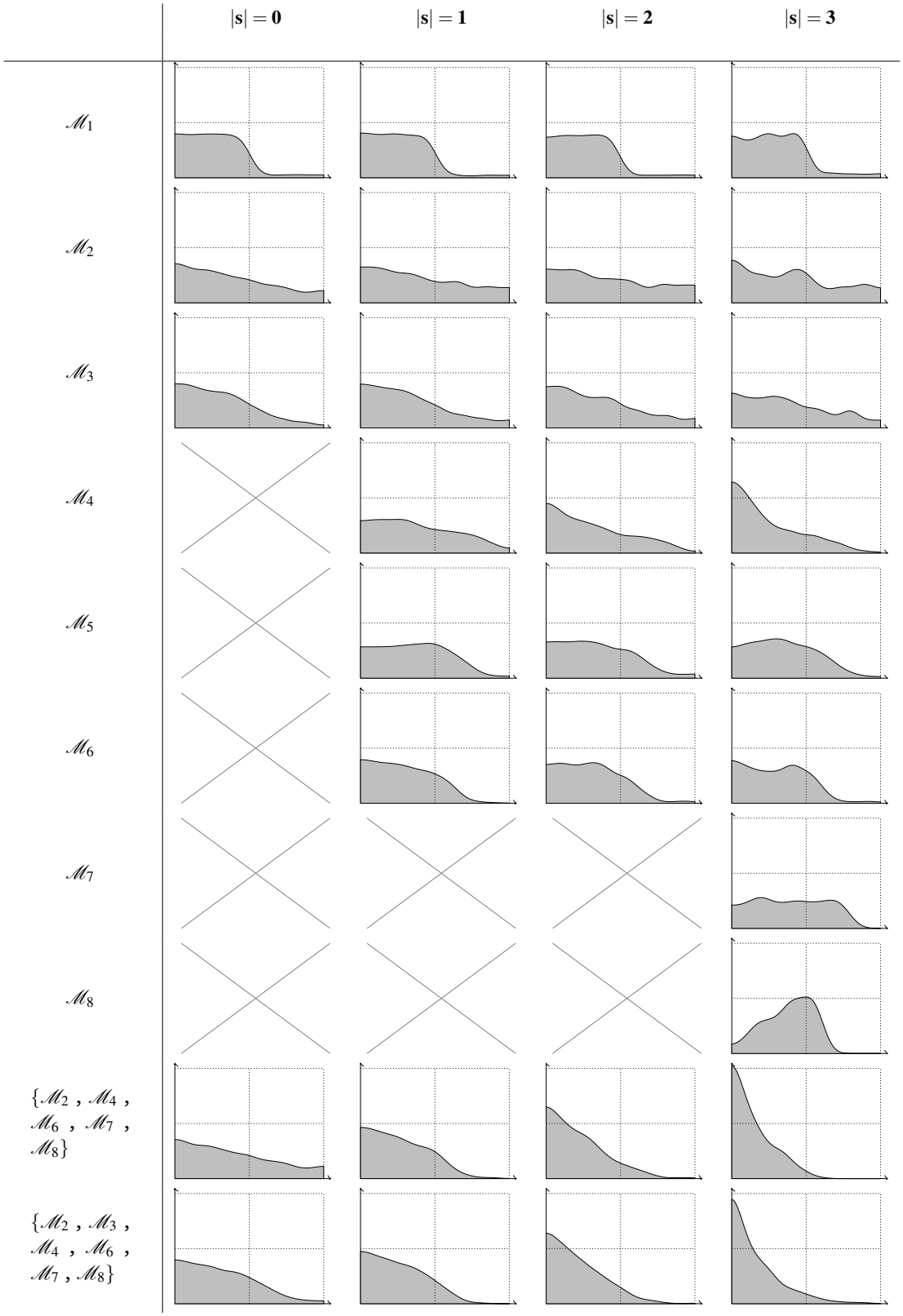


TABLE 5.1 – Courbes de la probabilité de tirer une donnée pertinente en fonction de son rang dans le tri (équation 5.1). Abscisse : rang r. Ordonnée : $pertinence\left(s \cup \{d_{(\mathbf{r})_s}\}\right)$. Les deux repères horizontaux sont à 0.25 et 0.5. Résultat obtenu sur un ensemble de couples d'images présentant des homographies plus ou moins marquées. Chaque ligne correspond à un ensemble différent de mesures utilisées. Les mesures sont décrites dans le tableau 5.3.

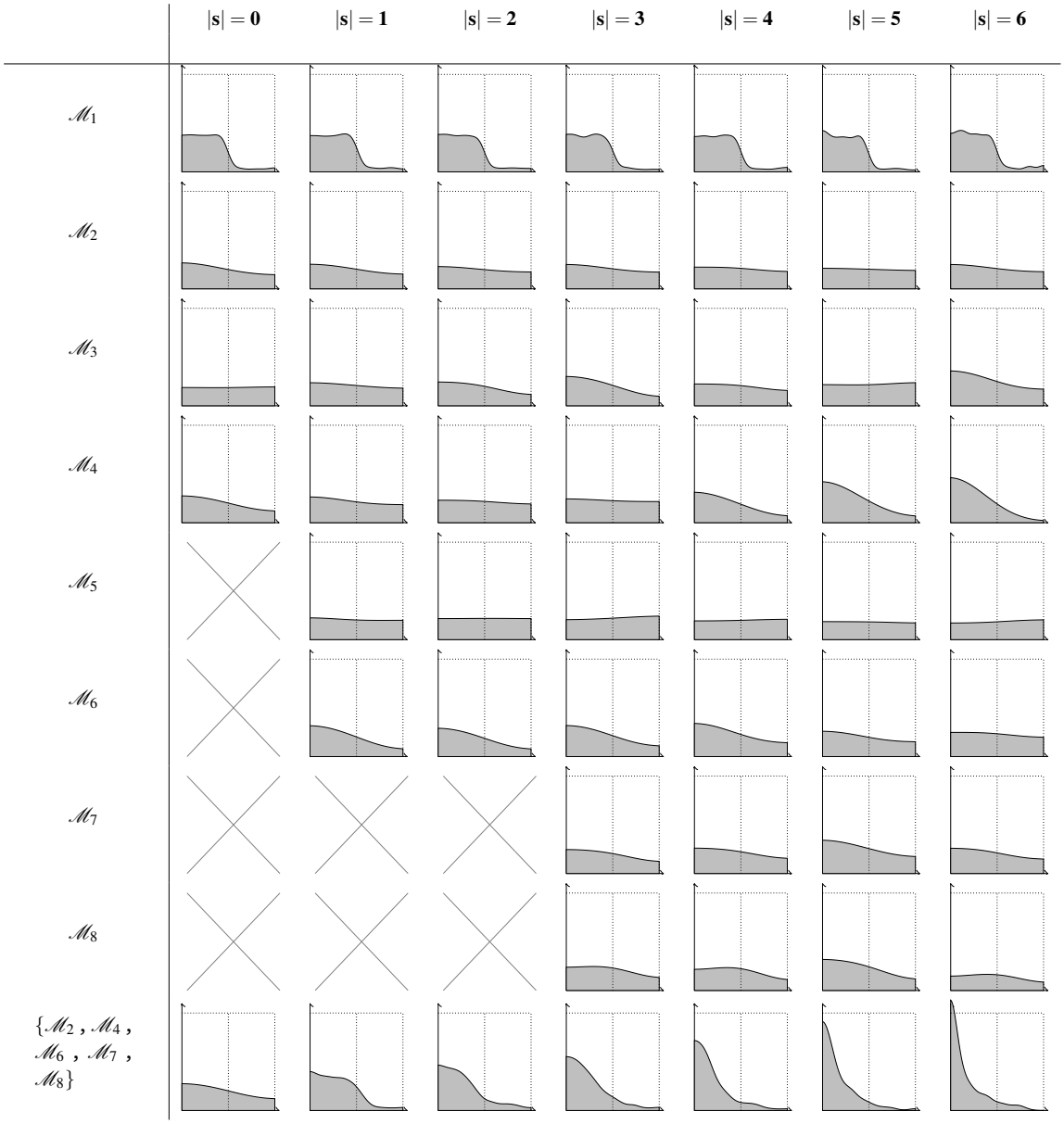


TABLE 5.2 – Courbes de la probabilité de tirer une donnée pertinente en fonction de son rang dans le tri (équation 5.1). Abscisse : rang r. Ordonnée : *pertinence* $\left(s \cup \{d_{(\mathbf{r})_s}\}\right)$. Résultat obtenu sur un ensemble de couples de photographies prises de points de vue plus ou moins différents. Chaque ligne correspond à un ensemble différent de mesures utilisées. Les mesures sont décrites dans le tableau 5.3.

Appellation		Description
$\mathscr{M}_1$	**Contamination (d'après V.T.)**	$\prod_{i=0}^{\lvert s\rvert-1} s_i \in \mathscr{D}_0$
$\mathscr{M}_2$	**Ratio SIFT**	$min\left(\{Match(s_i))\}_{s_i \in s}\right)$
$\mathscr{M}_3$	**Cohésion Affine**	$max\left(\{\lVert Aff(s_i) - Aff(s_k)\rVert\}_{s_i \in s}, s_k \in \mathscr{D}\right)$ avec, s_k le k-ième plus proche voisin de s_i dans l'espace des transformations affines. k est de l'ordre de $\frac{N}{50}$
$\mathscr{M}_4$	**Distance Minimale**	$min\left(\{\lVert P(s_i) - P(s_j)\rVert\}_{s_i,s_j \in s} \cup \{\lVert P'(s_i) - P'(s_j)\rVert\}_{s_i,s_j \in s}\right)$
$\mathscr{M}_5$	**Cohérence des Similitudes**	$max\left(\{\lVert Sim(s_i) - Sim(s_j)\rVert\}_{s_i,s_j \in s}\right)$
$\mathscr{M}_6$	**Cohérence Affine**	$max\left(\{\lVert Aff(s_i) - Aff(s_j)\rVert\}_{s_i,s_j \in s}\right)$
$\mathscr{M}_7$	**Ordre des Points**	L'ordre des 4 points du quadrilatère $(P(s_0), P(s_1), P(s_2), P(s_3))$ est conservé dans le quadrilatère $(P'(s_0), P'(s_1), P'(s_2), P'(s_3))$
$\mathscr{M}_8$	**Ratio Triangles**	$\lVert Q - Q'\rVert$ avec $Q = \frac{1}{\sum_{i=1}^{4} T_i}[T_1, T_2, T_3, T_4]$ et $Q' = \frac{1}{\sum_{i=1}^{4} T'_i}[T'_1, T'_2, T'_3, T'_4]$, où T_i et $T'_i, i = 1,2,3,4$ sont les aires des triangles formés par les correspondances dans la première et la seconde image

TABLE 5.3 – Description des mesures utilisées pour l'estimation des homographies et matrices fondamentales. Chaque mesure donne un score de *pertinence* (c.a.d. entre autre la cohérence de l'ensemble de données mesuré).

	Algorithme	Iter. Moy.	Iter. Min.	Iter. Max.	Temps (ms)	Accélér.
1	RANSAC	8181.9	39	40910	1595.5	1
	PROSAC	1709.9	31	7628	333.3	4.79
	BetaSAC avec $\mathscr{M}_2$	1548.6	24	8142	289.3	5.51
	BetaSAC avec $\mathscr{M}_6$	287.0	3	1840	55.51	**28.74**
	OABSAC	157.8	1	1131	30.3	**52.66**
2	RANSAC	15858.5	338	90974	1447.4	1
	PROSAC	6445.6	30	39564	587.5	2.46
	BetaSAC avec $\mathscr{M}_2$	6414.0	32	41183	602.3	2.4
	BetaSAC avec $\mathscr{M}_6$	636.9	1	6697	63.8	**22.69**
	OABSAC	372.2	1	5701	38.7	**37.40**
3	RANSAC	1302.8	9	6179	1293.0	1
	PROSAC	1533.6	129	3990	1523.1	0.85
	BetaSAC avec $\mathscr{M}_2$	678.2	4	2601	677.9	1.91
	BetaSAC avec $\mathscr{M}_6$	30.8	1	157	30.9	**41.85**
	OABSAC	13.5	1	96	16.0	**80.81**
4	RANSAC	687.2	1	4975	198.0	1
	PROSAC	329.7	34	1156	94.9	2.09
	BetaSAC avec $\mathscr{M}_2$	242.8	2	976	71.0	2.79
	BetaSAC avec $\mathscr{M}_6$	7.1	1	29	2.12	**93.41**
	OABSAC	6.0	1	29	1.97	**100.50**

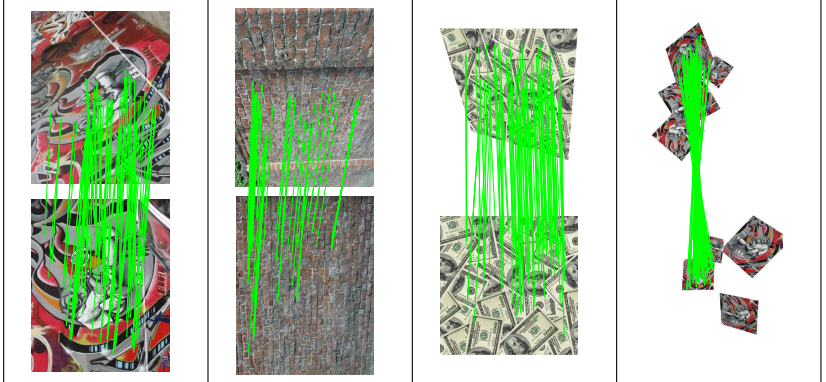
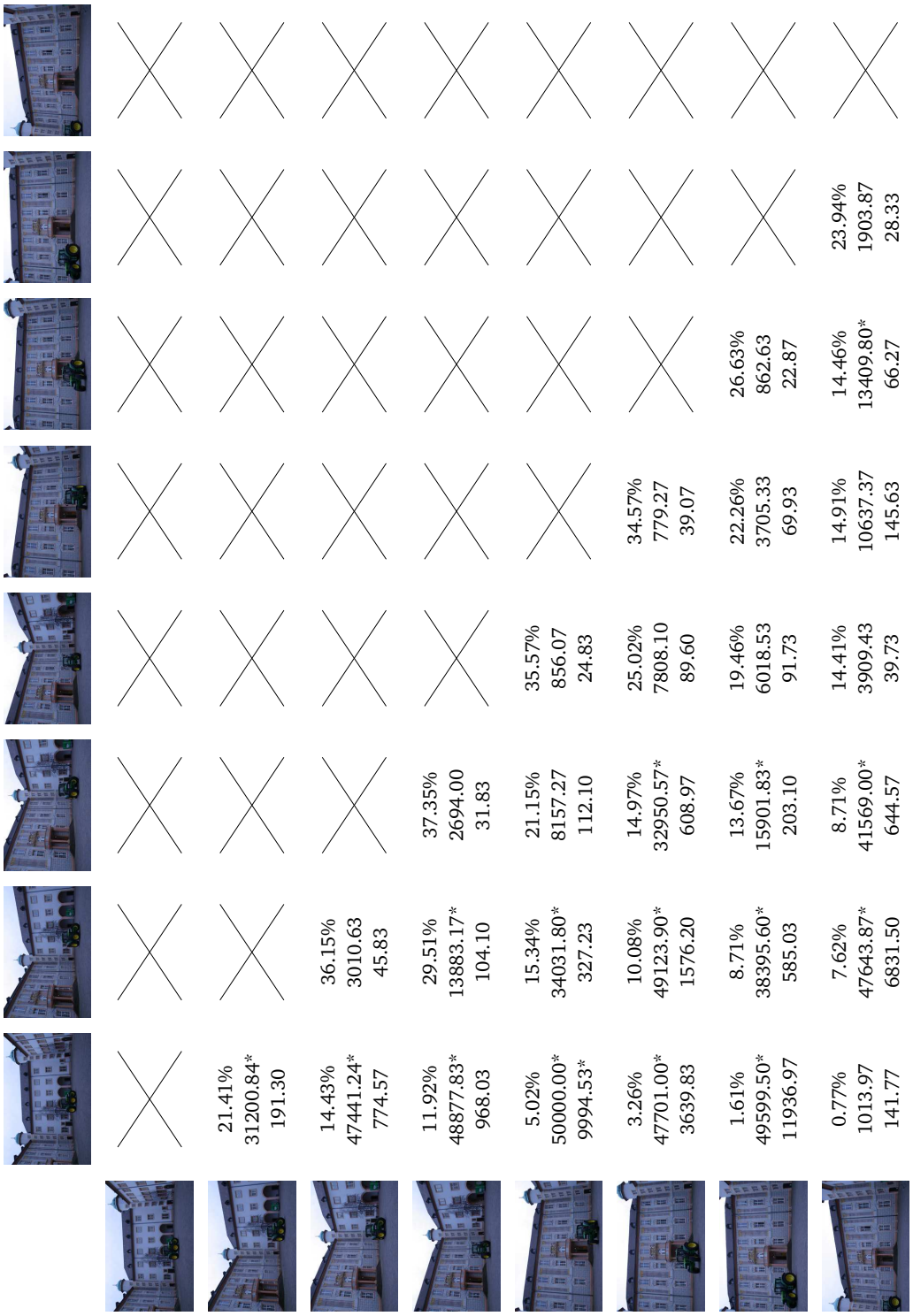


TABLE 5.4 – Résultats obtenus par RANSAC, PROSAC et nos méthodes BetaSAC et OABSAC pour différents problèmes d'estimation d'une homographie. Pour chaque méthode et couple d'images, nous mesurons le nombre d'itérations moyen, minimum, maximum ainsi que le temps moyen en millisecondes sur 100 tentatives pour résoudre le problème (trouver un modèle d'homographie rassemblant $\alpha = 75\%$ des correspondances. Les paramètres de BetaSAC utilisés sont $n = 10$ et $p = 3$. OABSAC est utilisé avec l'ensemble de mesures $\{\mathscr{M}_2, \mathscr{M}_3, \mathscr{M}_4, \mathscr{M}_6, \mathscr{M}_7, \mathscr{M}_8\}$ et sans adaptation en ligne. Le nombre d'itérations guidées T est défini à 200000. Les paires d'images 1 et 2 sont des homographies très prononcées. Les paires 3 et 4 sont des exemples semi-synthétiques montrant le bénéfice de nos méthodes en présence d'une répétition de motifs.



	21.41% 31200.84* 191.30								
	14.43% 47441.24* 774.57	36.15% 3010.63 45.83							
	11.92% 48877.83* 968.03	29.51% 13883.17* 104.10	37.35% 2694.00 31.83						
	5.02% 50000.00* 9994.53*	15.34% 34031.80* 327.23	21.15% 8157.27 112.10	35.57% 856.07 24.83					
	3.26% 47701.00* 3639.83	10.08% 49123.90* 1576.20	14.97% 32950.57* 608.97	25.02% 7808.10 89.60	34.57% 779.27 39.07				
	1.61% 49599.50* 11936.97	8.71% 38395.60* 585.03	13.67% 15901.83* 203.10	19.46% 6018.53 91.73	22.26% 3705.33 69.93	26.63% 862.63 22.87			
	0.77% 1013.97 141.77	7.62% 47643.87* 6831.50	8.71% 41569.00* 644.57	14.41% 3909.43 39.73	14.91% 10637.37 145.63	14.46% 13409.80* 66.27	23.94% 1903.87 28.33		

TABLE 5.5 – Résultats obtenus par OABSAC lors de la comparaison de différents couples d'images issues de la scène "castle" de [63]. Les trois informations résultant de chaque comparaison sont : le taux de correspondances correctes et le temps (en millisecondes) mis par RANSAC et OABSAC pour estimer la géométrie épipolaire. Un astérisque à côté d'un temps signifie qu'il est sous-évalué car certaines tentatives ont atteint le nombre maximum d'itérations (50000) sans parvenir à un résultat. Chaque valeur de temps est une moyenne sur 100 tentatives. L'ensemble des mesures utilisées par OABSAC est $\{\mathscr{M}_2, \mathscr{M}_3, \mathscr{M}_4, \mathscr{M}_6, \mathscr{M}_7, \mathscr{M}_8\}$

TABLE 5.6 – Résultats obtenus par OABSAC lors de la comparaison de différents couples d'images issues de la scène "Herzjesu" de [63]. Les trois informations résultant de chaque comparaison sont : le taux de correspondances correctes et le temps (en millisecondes) mis par RANSAC et OABSAC pour estimer la géométrie épipolaire. Un astérisque à côté d'un temps signifie qu'il est sous-évalué car certaines tentatives ont atteint le nombre maximum d'itérations (50000) sans parvenir à un résultat. Chaque valeur de temps est une moyenne sur 100 tentatives. L'ensemble des mesures utilisées par OABSAC est $\{\mathscr{M}_2, \mathscr{M}_3, \mathscr{M}_4, \mathscr{M}_6, \mathscr{M}_7, \mathscr{M}_8\}$

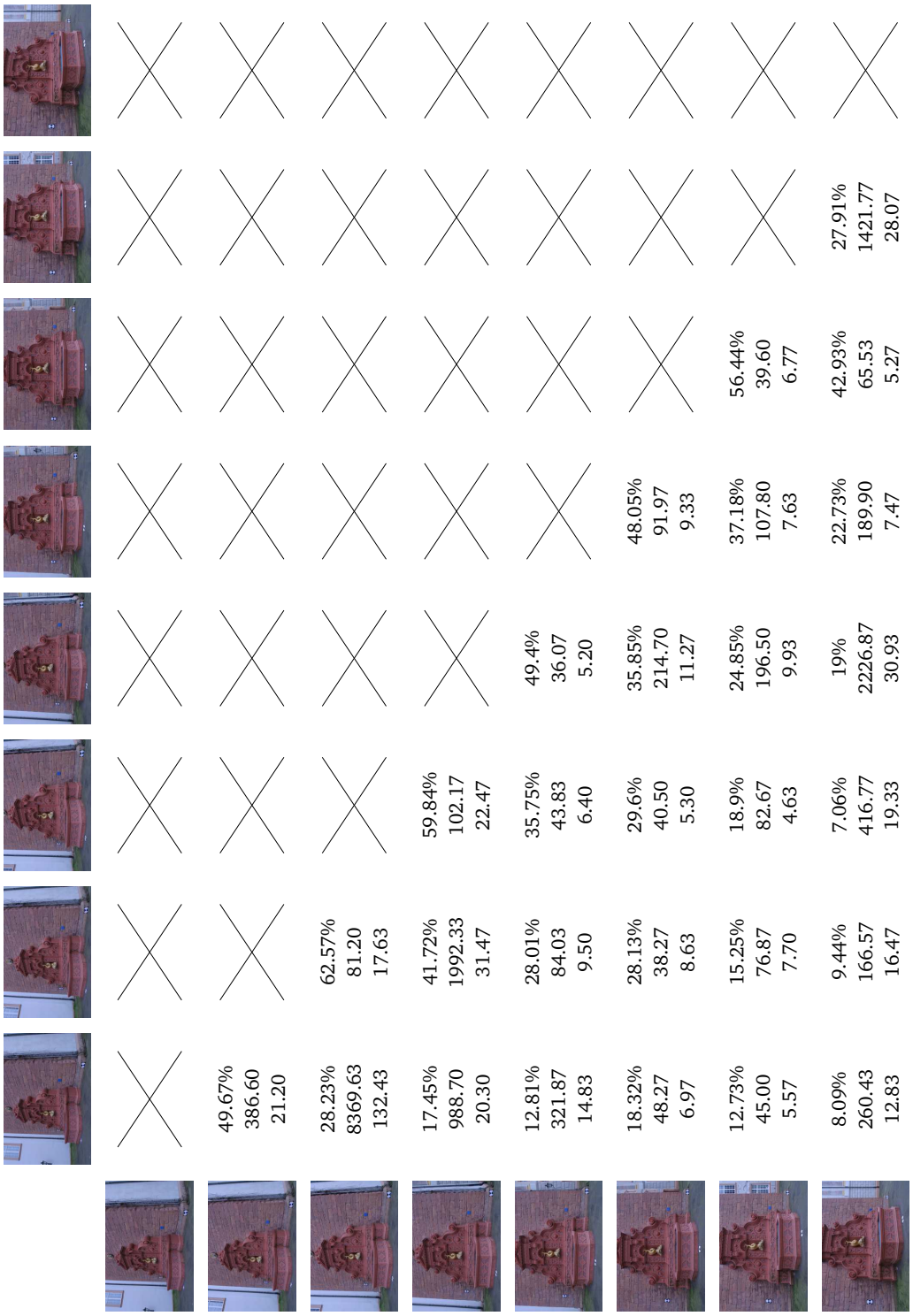


TABLE 5.7 – Résultats obtenus par OABSAC lors de la comparaison de différents couples d'images issues de la scène "fountain" de [63]. Les trois informations résultant de chaque comparaison sont : le taux de correspondances correctes et le temps (en millisecondes) mis par RANSAC et OABSAC pour estimer la géométrie épipolaire. Un astérisque à côté d'un temps signifie qu'il est sous-évalué car certaines tentatives ont atteint le nombre maximum d'itérations (50000) sans parvenir à un résultat. Chaque valeur de temps est une moyenne sur 100 tentatives. L'ensemble des mesures utilisées par OABSAC est $\{\mathscr{M}_2, \mathscr{M}_3, \mathscr{M}_4, \mathscr{M}_6, \mathscr{M}_7, \mathscr{M}_8\}$

6

Conclusions

6.1 Contributions

Ce travail s'inscrit dans le contexte de l'estimation robuste des paramètres d'un modèle à partir de données en partie erronées. Notre approche est de sélectionner des échantillons de données les plus pertinents possibles pour instancier des paramétrisations du modèle recherché les plus proches possibles de la vérité.
Pour ce faire, nous avons apporté ces différentes contributions :

1. Nous avons proposé une nomenclature pour distinguer et définir différents degrés de qualité d'un ensemble de données : la *non contamination*, la *cohésion*, la *cohérence* et enfin la *pertinence*.
2. Nous avons unifié l'ensemble des améliorations dites *par réordonnancement de l'échantillonnage* de RANSAC sous un formalisme commun.
3. Nous avons proposé BetaSAC, un algorithme de réordonnancement original, se focalisant sur la formation d'échantillons *pertinents* plutôt que simplement *non contaminés*, grâce à des tris conditionnels partiels des données. La pertinence peut être estimée à partir d'information provenant des images elles-mêmes ainsi que de sources de données complémentaires quelconques.
4. Nous en avons dérivé une version optimale et adaptative, OABSAC, faisant intervenir une phase d'apprentissage hors ligne. Cet apprentissage a pour but de mesurer les propriétés caractéristiques du problème spécifique que l'on souhaite résoudre, de façon à établir automatiquement le paramétrage optimal de notre algorithme. Ce paramétrage est celui qui doit mener à une estimation suffisamment précise des paramètres du modèle recherché en un temps (en secondes) le plus court.

6.2 Bilan

Les deux méthodes proposées, sont des solutions très générales, qui permettent d'intégrer dans RANSAC tout type d'information complémentaire utile à la résolution du problème. Nous avons montré, sur les problèmes de l'estimation d'une homographie ou d'une géométrie épipolaire entre deux images, que notre tri conditionnel des données permet de satisfaire des propriétés de cohérence des échantillons formés, menant à une génération d'échantillons pertinents dans les premières itérations de BetaSAC, et donc à une résolution rapide du problème. Pour ce faire, nous avons utilisé plusieurs indices de cohérence, issus du signal (déformation affine locale), de la mise en correspondance des points d'intérêt (score de similarité d'apparence), de critères de précision (distance minimale entre deux points d'intérêt) ou encore de critères géométriques (non symétrisation de l'ordre des points d'intérêt, respect des ratios des aires des triangles). La plupart de ces indices ne demandent aucun calcul supplémentaire ni matériel spécifique, car nous utilisons une information qui est déjà présente mais sous-exploitée par les méthodes précédentes. De plus, la partialité des tris effectués assure la rapidité et la randomisation, indispensable à ce type de méthodes.

Notre étude expérimentale montre des gains de vitesse de résolution d'estimation des paramètres atteignant un facteur 100 par rapport à l'algorithme RANSAC classique et un facteur 10 par rapport à la méthode de l'état de l'art, PROSAC, pour les deux applications que nous considérons.

6.3 Perspectives

Une des pistes, sans toute parmi des plus prometteuses pour notre méthode, est l'exploitation des vidéos, en particulier pour des applications de suivi d'objet ou de SLAM (Simultaneous Localization And Mapping). En effet, une vidéo contient évidemment beaucoup plus d'information exploitable qu'une image fixe, on peut donc espérer accélérer son traitement d'autant plus. On peut en effet penser à de nombreux indices de cohérence (issus de la cohérence temporelle) et *aprioris* spécifiques, provenant des frames précédentes.

Pour les cas les plus difficiles, nécessitant beaucoup d'itérations, les itérations passées constituent également une source d'information riche. Quelques recherches s'y sont déjà intéressé, mais aucune solution offre les même garanties que les méthodes par réordonnancement. Or, l'utilisation de cette information sans ces garanties fait courir le risque de perdre la randomisation indispensable à ce type de méthodes.

Nous pensons aussi que des efforts doivent être mis en œvre dans l'étude d'un critère d'arrêt exploitant mieux la précocité des méthodes par réordonnancement de l'échantillonnage. En effet, posséder un échantillonnage efficace ne sert à rien si l'algorithme n'est pas capable de s'arrêter suffisamment tôt. A notre connaissance, un tel critère, accompagné d'une justification rigoureuse n'existe pas encore.

Nous voulons également mesurer ce que peuvent apporter des capteurs peu coûteux tels qu'un accéléromètre, un gyroscope, un GPS ou encore une grille de lasers mesurant la profondeur (type Kinect), à l'estimation du mouvement d'une caméra.

7

Annexe

7.1 Reformulation de notre problème

7.1.1 Formations de correspondances entres points de deux images

La formation de correspondances entre deux images se fait en trois temps :

1. La détection de points d'intérêt sur les deux images.
2. Le calcul d'un vecteur de description en chaque point d'intérêt. Ce vecteur contient l'information visuelle locale.
3. La mise en correspondance des points d'intérêt possédant une description proche. Les points mis en correspondance ont des chances d'être les mêmes points d'un même objet sur les deux images.

Cette section décrit les techniques que nous avons utilisé pour réaliser chacune de ces étapes.

7.1.1.1 Détection de points d'intérêt

On ne peut pas mettre tous les points d'une image en correspondance avec les points de la seconde image car le nombre de correspondances et le taux d'erreurs seraient bien trop élevés. Alors, on ne forme des correspondances qu'entre des *points d'intérêt* des images.

Un point d'intérêt idéal doit posséder ces propriétés :

– *Répétabilité* : si deux photographies d'un même objet sont prises dans des conditions différentes, alors un bon pourcentage de caractéristiques détectées sur la première photo doivent aussi l'être sur la seconde.
– *Information* : l'intensité autour du point d'intérêt doit montrer des variations importantes de façon à être caractéristique et distinguable.
– *Localité* : les caractéristiques doivent être locales pour limiter l'influence des informations non pertinentes (fond, occultation par un autre objet, ...). La localité permet aussi une approximation simple des déformations géométriques et photométriques entre deux images d'un même objet.
– *Quantité* : le nombre de caractéristiques détectées doit être suffisamment grand pour que l'on en trouve assez même sur les objets les plus petits.
– *Précision* : les caractéristiques doivent être localisées précisément tant dans l'espace que dans la dimension d'échelle.
– *Efficacité* : la détection doit être rapide.

Bien sûr l'importance de ces propriétés dépend de l'application. D'ailleurs certaines d'entre elles sont concurrentes. Par exemple, plus la caractéristique est locale, moins elle peut contenir d'information et inversement. De la même façon, l'augmentation de la répétabilité va souvent nécessiter une plus grande résistance aux déformations et donc une diminution de l'information.

Exemple du détecteur de Harris multi-échelles Des travaux sur les points d'intérêt [8] ont montré que le détecteur de Harris était le plus répétable. Cependant, cette répétabilité se dégrade très vite avec

un changement d'échelle, il faut donc l'adapter au changement d'échelle [65]. L'idée maîtresse de ce détecteur est d'utiliser la fonction d'auto-corrélation pour déterminer les positions où le signal change dans les deux directions simultanément. Notons $L_x(\sigma)$ et $L_y(\sigma)$ les dérivées premières du signal en x et y sur une fenêtre gaussienne de rayon σ, et $G(\tilde{\sigma})$ la fonction gaussienne de variance $\tilde{\sigma}$. Une matrice C liée à la fonction d'auto-corrélation est calculée :

$$C(x,\sigma,\tilde{\sigma}) = G((x,\tilde{\sigma}) \otimes \begin{bmatrix} {L_x}^2(x,\sigma) & L_xL_y(x,\sigma) \\ L_xL_y(x,\sigma) & {L_y}^2(x,\sigma) \end{bmatrix}$$

Les vecteurs propres de cette matrice sont les courbures principales de la fonction d'auto-corrélation. Deux valeurs significatives indiquent la présence d'un point d'intérêt. En pratique, on retient un point comme point d'intérêt si on a $v(x,\sigma,\tilde{\sigma}) = det(C) - \alpha trace^2(C) > seuil_h$ ou $seuil_h$ est un seuil à fixer en fonction du nombre de points que l'on veut obtenir et α est un paramètre fixé expérimentalement. On exige aussi que cette valeur atteigne un maximum local en x. L'invariance à la rotation est assurée par la symétrie de la matrice C.

Mais en présence d'un changement d'échelle s entre deux images , le détecteur de points d'intérêt doit être adapté pour obtenir des résultats répétables. Prenons l'exemple simple d'une image de carré :

Points détectés sur un carré à l'échelle 1

Point détecté sur le même carré à léchelle s=1/3

On voit que le résultat de la détection de points d'intérêt dépend de l'échelle. Cela vient du fait que la notion de dérivation d'un signal discret, donc de "coin", est relative à une échelle. Pour retrouver les mêmes points sur la deuxième image il faut se placer à une échelle inférieure. Pour cela, il faut adapter la matrice ainsi :

$$s^2G(x,s\tilde{\sigma}) \otimes \begin{bmatrix} {L_x}^2(x,s\sigma) & L_xL_y(x,s\sigma) \\ L_xL_y(x,s\sigma) & {L_y}^2(x,s\sigma) \end{bmatrix}, \text{ ou } s \text{ est l'échelle.}$$

Mais en général on ne connaît pas a priori le changement d'échelle entre deux images d'un même objet. La détection des points d'intérêt se fait alors à plusieurs échelles $s_n = s_0 s^n$. s_0 est l'échelle initiale correspondant à la plus haute résolution de l'image et s_n dénote les niveaux consécutifs dans l'espace d'échelle.

Opérateur laplacien Il n'est pas nécessaire de conserver tous les points détectés à toutes les échelles. J. L. Crowley propose [15] de ne conserver un point trouvé dans une certaine échelle de représentation de l'image que si elle est l'échelle caractéristique du voisinage de ce point. Ainsi, un point conservé dans la première image à l'échelle s_1 devrait être conservé dans la seconde I^2 à l'échelle $s_2 = s * s_1$, si l'échelle caractéristique est un bon invariant. Une façon simple de déterminer si un point est positionné sur une échelle caractéristique est de regarder si sa valeur $v(x,\sigma,\tilde{\sigma})$ atteint un maximum local dans la dimension des échelles. Mais trop peu de points sont à la fois un maximum dans les deux dimensions d'espace et dans la dimension d'échelle. Alors on construit une autre représentation en échelles avec le laplacien :

$$Laplacien(x,\sigma) = \mid s_n^2(L(x,s_n\sigma) + L_{yy}(x,s_n\sigma)) \mid$$

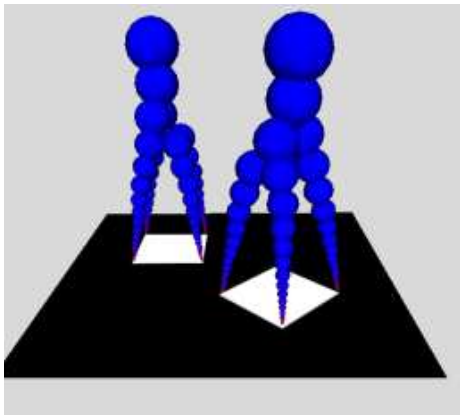
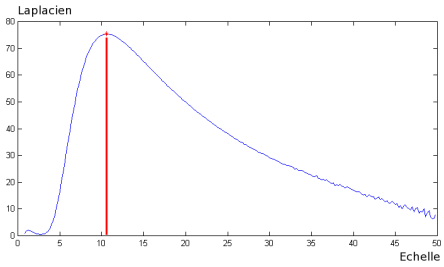


FIGURE 7.1 – Points d'intérêt détectés à toutes les échelles sur des carrés. La taille des sphères figure l'échelle.

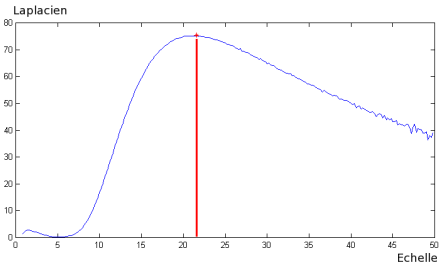
Un point est alors conservé si il est un maximum local de $v(x,\sigma,\tilde{\sigma})$ dans les deux dimensions d'espace et s'il est un maximum local du Laplacien dans la dimension d'échelle. Ce détecteur est appelé le détecteur de Harris-Laplacien.

Photo à l'échelle 1

Photo de la même scène avec un zoom $\times 2$

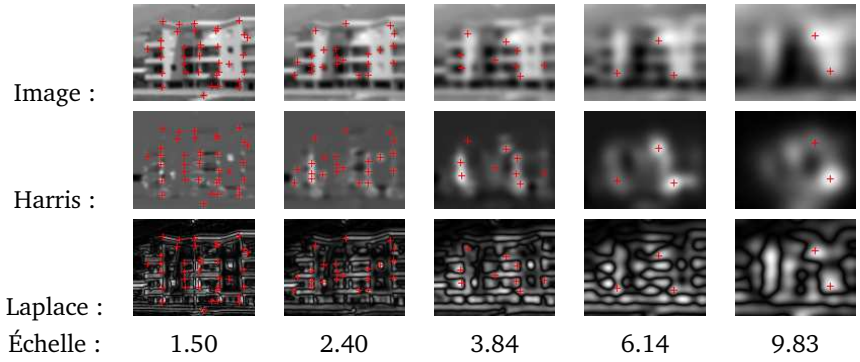


Sur le point d'intérêt marqué, le laplacien atteint son maximum à l'échelle 10.6



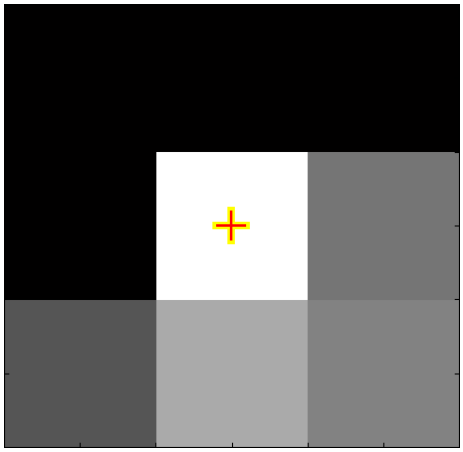
Au même point, le laplacien atteint son maximum à l'échelle 21.6

On voit sur ces deux courbes que le laplacien est une bonne fonction d'échelle car il n'atteint qu'un seul maximum local (ce qui n'est toutefois pas toujours le cas) et le rapport des deux échelles du maximum est bien égal au facteur d'échelle entre les deux images.

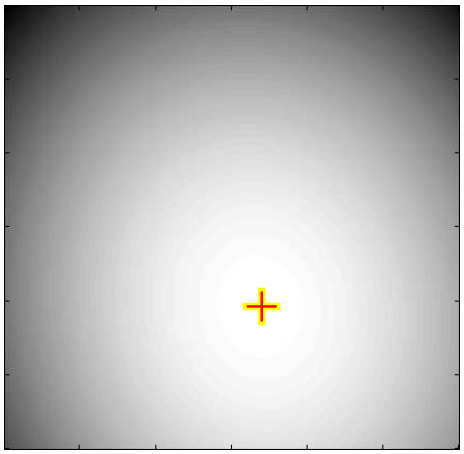


Points d'intérêt détectés sur les différentes échelles de l'image par Harris-Laplace

Affinage de la position des points dans l'espace et l'échelle Pour la suite, il est important d'avoir une très bonne précision sur la position des points d'intérêt. La résolution pixellique n'est donc pas suffisante. Pour obtenir une meilleure précision nous approchons aux moindres carrés le voisinage des points trouvés dans le domaine de Harris par une surface polynomiale d'ordre 2. Ainsi, on place le point non plus au centre du pixel mais sur le maximum local de la surface polynomiale.



Voisinage 3x3 d'un maximum local dans le domaine de Harris



Approximation quadratique du voisinage et repositionnement du point d'intérêt

Obtenir une valeur d'échelle précise est tout aussi important pour la suite. Alors, pour l'affiner nous cherchons le maximum de la courbe polynomiale de degré deux qui interpole les trois valeurs du laplacien autour de l'échelle caractéristique discrète trouvée précédemment

Normalisation affine

7.1.1.2 Description des points d'intérêt

Les caractéristiques détectées sont inutiles sans un bon descripteur. C'est lui qui est capable de capturer l'information autour des points détectés. Un bon descripteur doit posséder ces quatre propriétés :

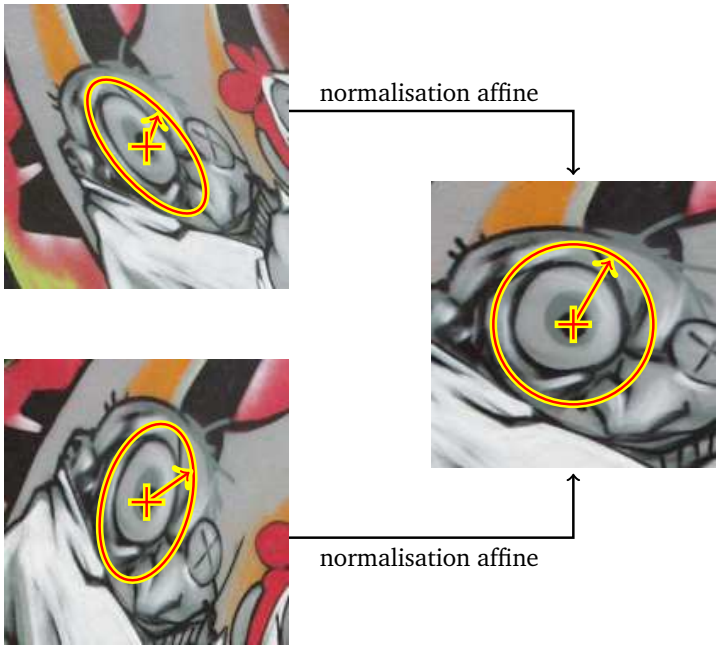


FIGURE 7.2 – Illustration de la normalisation affine effectuée autour d'un point d'intérêt de manière à rendre l'apparence localement invariante à un changement de point de vue.

- *Répétabilité* : si deux points d'intérêt, de deux images différentes, sont positionnés sur le même point du même objet, alors leurs descriptions doivent être aussi proches que possible.
- *Information* : une description doit contenir assez d'information pour distinguer deux points différents.
- *Compacité* : une description ne doit pas contenir de redondance ou d'information fortement corrélée. Ainsi les applications seront plus efficaces tant en terme de mémoire que de temps de calcul.
- *Efficacité* : le nombre de caractéristiques locales détectées pouvant être important (plusieurs milliers), le calcul des descriptions doit être rapide.

Là encore, certaines propriétés se font concurrence (comme la compacité et l'efficacité par exemple). Il n'existe donc pas un bon descripteur mais des bons descripteurs, en fonction de l'application.

Le descripteur que nous avons utilisé est le descripteur SIFT [29]. Ce descripteur n'est pas invariant par rotation. Alors la première étape est de trouver l'orientation à laquelle il va être calculée. David Lowe propose de le calculer selon la direction principale du gradient dans la localité du point d'intérêt. Dans le cas ou plusieurs directions dominantes ressortent (pensons par exemple au coin d'un carré), plusieurs descriptions sont calculées. Il peut ainsi y avoir jusqu'à 4 signatures SIFT par point d'intérêt.

Une fois que l'on a une direction et une échelle (celle associée au point d'intérêt), on peut se donner une fenêtre carrée sur laquelle on va calculer la description. Le gradient (module et orientation) est alors calculé en chaque point de la fenêtre en utilisant les convolutions par les dérivées de gaussiennes déjà réalisées dans la phase de détection des points d'intérêt. La fenêtre est partagée en 4x4 cases, contenant chacune un histogramme de 8 secteurs angulaires. Chaque gradient calculé dans une de ces cases contribue à l'histogramme proportionnellement à son module. Une fonction de poids gaussienne est ajoutée au centre de la fenêtre pour que l'influence du gradient tombe avec l'éloignement au point d'intérêt. Et chacune des dimensions de l'histogramme 3D ainsi calculée est lissée pour rendre le descripteur moins perturbable par un petit changement de position ou d'orientation. La signature du point est codée dans un vecteur de $4*4*8 = 128$ valeurs. Ce vecteur est normalisé pour le rendre invariant aux variations affines de l'illumination, ainsi que tronqué à 20% pour limiter les effets d'un changement non linéaire de celle-ci. Il existe d'autres paramétrages du SIFT mais celui-ci est un bon compromis entre la dimension

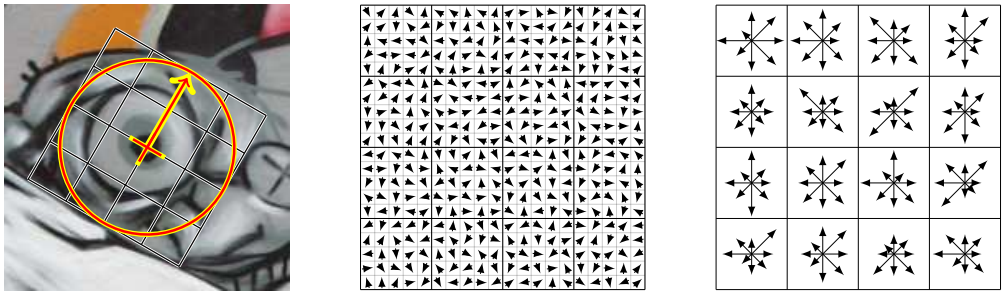


FIGURE 7.3 – Un descripteur SIFT formé de 16 histogrammes angulaires

de la signature et les performances du descripteur.

7.1.1.3 Mise en correspondance

Pour mettre les points d'intérêt des deux images en correspondance, nous calculons la distance entre les descriptions des points de la première image et celle des points de la seconde. La distance utilisée est simplement la distance euclidienne entre les vecteurs de description. Deux points sont mis en correspondance si le second point est le plus proche du premier et si il n'existe pas d'autre point presque aussi proche.

Information associée à chaque correspondance A chaque correspondance retenue, on peut associer ces deux informations complémentaires :

- Le score de mise en correspondance.
- La transformation de type similitude ou affine.

Cet ensemble de correspondances constitue les données d'entrée de notre algorithme.

7.1.2 Modèle recherché

Notre thechnique trouve trois applications principales : la création de panoramas, la reconnaissance et détection d'objets plans, l'estimation de la géométrie épipolaire entre deux images. Ces deux premières applications sont très similaires, elles utilisent toute deux un modèle homographique. La troisième, plus compliqué, utilise un modèle de géométrie épipolaire.

7.1.2.1 Transformation homographique

Une homographie, ou transformation projective, est l'application qui transforme un carré en quadrilatère quelconque. C'est la transformation subie par un objet plan, entre deux photographies dont le point de vue ou l'orientation diffèrent. C'est également la transformation subie par une scène rigide quelconque, entre deux photographies d'orientation différente mais de même point de vue. L'homographie n'est pas une transformation linéaire, mais elle peut néanmoins se caractériser par une matrice de taille 3×3, grâce aux coordonnées homogènes. Cette matrice est invariante à un facteur multiplicatif près, on peut donc fixer sa dernière valeur à 1. L'homographie est donc une transformation à 8 paramètres, nécessitant au moins 4 correspondances entre les points de deux images pour être estimée.

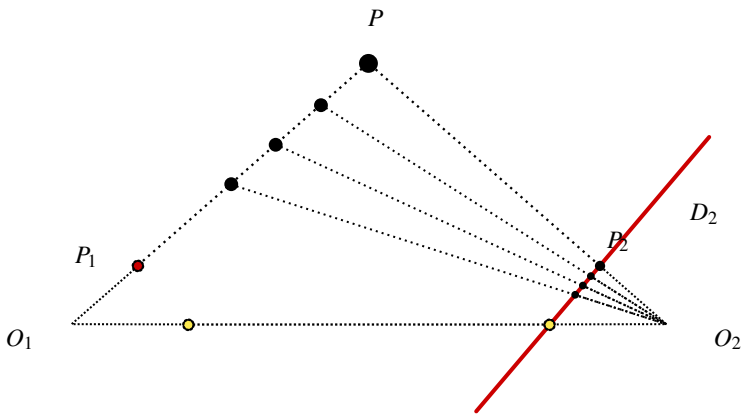


FIGURE 7.4 – Le point P se projette respectivement en P_1 et P_2 sur l'image de gauche et de droite. La matrice fondamentale F de l'image de gauche vers l'image de droite ne permet pas de trouver P_2 à partir de P_1. En revanche, elle permet de trouver la droite D_2, sur laquelle se trouve P_2, grâce à l'identité : $D_2 = F \cdot P_1$

$$P_2 = H \cdot P_1$$

$$\begin{bmatrix} P_{2x} \\ P_{2y} \\ w_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} P_{1x} \\ P_{1y} \\ w_1 \end{bmatrix} \tag{7.1}$$

Équation liant un point P_1 à son image P_2 dans le cas d'une transformation homographique

7.1.2.2 Géométrie épipolaire

La géométrie épipolaire est une contrainte traduisant le fait que lorsqu'on lance un rayon depuis le centre optique du premier point de vue, ce rayon est un point sur la première image et une droite sur la seconde. Là encore, grâce aux coordonnées homogènes, on peut définir une géométrie épipolaire par une matrice de taille 3×3. Cette matrice, appelée matrice épipolaire, permet de passer du point d'une image à la droite correspondante sur l'autre image. Pour estimer la matrice épipolaire entre deux images, au moins 7 ou 8 correspondances sont nécessaires.

$$D_2 = F \cdot P_1$$

$$\begin{bmatrix} D_{2a} \\ D_{2b} \\ D_{2c} \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} P_{1x} \\ P_{1y} \\ w_1 \end{bmatrix} \tag{7.2}$$

Équation liant un point P_1 à la droite D_2 associée dans le cas d'un objet rigide quelconque

Bibliographie

[1] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. pages 714–725, 1987.

[2] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces : Recognition using class specific linear projection, 1997.

[3] P.N. Belhumeur and D.J. Kriegman. What is the set of images of an object under all possible lighting conditions ? In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 270 –277, jun 1996.

[4] T. Botterill, S. Mills, and R. Green. New conditional sampling strategies for speeded-up ransac. In *BMVC09*, 2009.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] M. Brown, Gang Hua, and S. Winder. Discriminative learning of local image descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1) :43 –57, jan. 2011.

[7] Eric Bughin and Andrés Almansa. Planar Patch Detection for Disparity Maps. In *Proc. 3DPVT*, Espace Saint Martin, May 2010.

[8] R. Mohr C. Schmid and C. Bauckhage. Comparing and evaluating interest points. January 1998.

[9] Chum Matas Center, O. Chum, and J. Matas. Randomized ransac with t d,d test. In *Image and Vision Computing*, pages 448–457, 2002.

[10] Sunglok Choi and Jong-Hwan Kim. Robust regression to varying data distribution and its application to landmark-based localization. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 3465 –3470, oct. 2008.

[11] Ondrej Chum and Jiri Matas. Matching with prosac " progressive sample consensus. In *CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 220–226, Washington, DC, USA, 2005. IEEE Computer Society.

[12] Ondrej Chum, Jiri Matas, and Josef Kittler. Locally optimized ransac. In *DAGM-Symposium*, pages 236–243, 2003.

[13] Ondřej Chum, Jiří Matas, and Štěpán Obdržálek. Epipolar geometry from three correspondences. In Ondřej Drbohlav, editor, *Computer Vision — CVWW'03 : Proceedings of the 8th Computer Vision Winter Workshop*, pages 83–88, Prague, Czech Republic, February 2003. Czech Pattern Recognition Society.

[14] Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 772–779, Washington, DC, USA, 2005. IEEE Computer Society.

[15] James L Crowley. A representation for visual information. Technical Report CMU-RI-TR-82-07, Robotics Institute, Pittsburgh, PA, November 1981.

[16] James L. Crowley and Olivier Riff. Fast computation of scale normalised gaussian receptive fields. In *Proceedings of the 4th international conference on Scale space methods in computer vision*, Scale Space'03, pages 584–598, Berlin, Heidelberg, 2003. Springer-Verlag.

[17] S. Teller E. Olson, M. Walter and J. Leonard. Single-cluster spectral graph partitioning for robotics applications. In *Proceedings of Robotics : Science and Systems (RSS)*, Cambridge, MA, USA, June 2005.

[18] C. L. Feng and Y. S. Hung. A robust method for estimating the fundamental matrix. In *In International Conference on Digital Image Computing*, pages 633–642, 2003.

[19] Martin A. Fischler and Robert C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981.

[20] Jan-Michael Frahm and Marc Pollefeys. Ransac for (quasi-)degenerate data (qdegsac). In *CVPR '06 : Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 453–460, Washington, DC, USA, 2006. IEEE Computer Society.

[21] L. Goshen and I. Shimshoni. Guided sampling via weak motion models and outlier sample generation for epipolar geometry estimation. pages I : 1105–1112, 2005.

[22] T. Tuytelaars H. Bay and L. van Gool. Surf : Speeded up robust features. 1978.

[23] C. Harris and M. Stephens. A combined corner and edge detector. 1988.

[24] Josef Sivic Andrew Zisserman James Philbin, Michael Isard. Descriptor learning for efficient retrieval. In *ECCV (3)*, pages 677–691, 2010.

[25] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector, 2004.

[26] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *Computers, IEEE Transactions on*, 42(3) :300 –311, mar 1993.

[27] Stephane Laveau and Olivier Faugeras. Oriented projective geometry for computer vision. In *ECCV96*, pages 147–156. Springer-Verlag, 1996.

[28] T. Lindeberg. Scale-space theory in computer vision. 1994.

[29] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[30] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[31] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2) :91–110, 2004.

[32] J Matas, O Chum, M Urban, and T Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10) :761 – 767, 2004. ¡ice :title¿British Machine Vision Computing 2002¡/ce :title¿.

[33] Jiri Matas and Ondrej Chum. Randomized ransac with sequential probability ratio test. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, pages 1727–1732, Washington, DC, USA, 2005. IEEE Computer Society.

[34] Jiri Matas and Ondrej Chum. Randomized ransac with sequential probability ratio test. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, pages 1727–1732, Washington, DC, USA, 2005. IEEE Computer Society.

[35] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1) :63–86, 2004.

[36] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1) :63–86, 2004.

[37] L. Moisan and B. Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57(3) :201–218, 2004.

[38] Joseph L. Mundy and Andrew Zisserman, editors. *Geometric invariance in computer vision*. MIT Press, Cambridge, MA, USA, 1992.

[39] Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3-d objects from appearance. *Int. J. Comput. Vision*, 14(1) :5–24, January 1995.

[40] D. R. Myatt, Philip H. S. Torr, Slawomir J. Nasuto, J. Mark Bishop, and R. Craddock. Napsac : High noise, high dimensional robust estimation - it's in the bag. In Paul L. Rosin and A. David Marshall, editors, *BMVC*. British Machine Vision Association, 2002.

[41] D.R. Myatt, P.H.S. Torr, S.J. Nasuto, J.M. Bishop, and R. Craddock. Napsac : High noise, high dimensional robust estimation - it's in the bag. In *BMVC02*, page Computer Vision Tools, 2002.

[42] Kai Ni, Hailin Jin, and Frank Dellaert. Groupsac : Efficient consensus in the presence of groupings. In *ICCV09*, Kyoto ;Japan, October 2009.

[43] Clark F. Olson. A general method for geometric feature matching and model extraction. *Int. J. Comput. Vision*, 45(1) :39–54, 2001.

[44] Tomas Pajdla, Tomas Werner, and Vaclav Hlavac. Oriented projective reconstruction, 1998.

[45] Michal Perďoch, Jiří Matas, and Ondřej Chum. Epipolar geometry from two correspondences. In Bob Werner, editor, *ICPR 2006 : Proceedings of the 18th International Conference on Pattern Recognition*, volume 4, pages 215–220, Los Alamitos, USA, August 2006. IEEE Computer Society.

[46] Massimiliano Pontil and Alessandro Verri. Support vector machines for 3d object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(6) :637–646, June 1998.

[47] Julien Rabin, Julie Delon, Yann Gousseau, and Lionel Moisan. MAC-RANSAC : a robust algorithm for the recognition of multiple objects. In *Proceedings of 3DPTV 2010*, page 051, Paris, France, 2010.

[48] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *In European Conference on Computer Vision*, pages 430–443, 2006.

[49] D. Roth, M. Yang, and N. Ahuja. Learning to recognize objects. In *CVPR*, pages 724–731, 2000.

[50] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4(3) :519–524, Mar 1987.

[51] S. Edelman T. Poggio. A network that learns to recognize 3d objects. *Nature*, 1991.

[52] Michael E. Tipping, Anita Faul, J J Thomson Avenue, and J J Thomson Avenue. Fast marginal likelihood maximisation for sparse bayesian models. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, pages 3–6, 2003.

[53] Roberto Toldo and Andrea Fusiello. Real-time incremental j-linkage for robust multiple structures estimation.

[54] G. Tolias and Y. Avrithis. Speeded-up, relaxed spatial matching. In *in Proceedings of International Conference on Computer Vision (ICCV 2011)*, Barcelona, Spain, November 2011.

[55] Ben J. Tordoff and David W. Murray. Guided-mlesac : Faster image transform estimation by using matching priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10) :1523–1535, 2005.

[56] P. Torr and A. Zisserman. Mlesac : A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78 :138–156, 2000.

[57] P.H.S. Torr and A. Zisserman. Mlesac : A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1) :138 – 156, 2000.

[58] Leonardo Trujillo and Gustavo Olague. Automated design of image operators that detect interest points. *Evol. Comput.*, 16(4) :483–507, December 2008.

[59] Andreas Turina, Tinne Tuytelaars, and Luc Van Gool. Efficient grouping under perspective skew, 2001.

[60] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1) :71–86, January 1991.

[61] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *Int. J. Comput. Vision*, 59(1) :61–85, August 2004.

[62] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(4) :376 –380, apr 1991.

[63] T. Werner and A. Zisserman. Model selection for automated architectural reconstruction from multiple views. In *British Machine Vision Conference*, pages 53–62, 2002.

[64] Tomas Werner and Tomas Pajdla. Oriented matching constraints, 2001.

[65] C. Schmid et R. Horaud Y. Dufournaud. Appariement d'images à des échelles différentes. 2000.

[66] Lihi Zelnik-Manor and Michal Irani. Multiview constraints on homographies. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(2) :214–223, 2002.