



**HAL**  
open science

# Auto-étalonnage de caméras fisheyes en environnement structuré

Clement Deymier, Thierry Chateau, Lébraly Pierre

► **To cite this version:**

Clement Deymier, Thierry Chateau, Lébraly Pierre. Auto-étalonnage de caméras fisheyes en environnement structuré. RFIA 2012 (Reconnaissance des Formes et Intelligence Artificielle), Jan 2012, Lyon, France. pp.978-2-9539515-2-3. hal-00656511

**HAL Id: hal-00656511**

**<https://hal.science/hal-00656511>**

Submitted on 17 Jan 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Auto-étalonnage de caméras *fisheyes* en environnement structuré

Clément Deymier<sup>1</sup>

Pierre Lébraly<sup>1</sup>

Thierry Chateau<sup>1</sup>

<sup>1</sup> Lasmea-UMR-CNRS 6602.

Campus des Cézeaux, 24 Avenue des Landais, 63177 Aubière Cedex - France.  
prénom.nom@lasmea.univ-bpclermont.fr

## Résumé

Cet article présente une méthode simple et totalement automatique de correction de la distorsion radiale des caméras à courte focale non étalonnées. Le principe consiste à estimer les paramètres intrinsèques d'un modèle de projection à partir de l'analyse d'une scène présentant des éléments tridimensionnels de type droite. Comme introduit par [3, 10], la projection des contours de ces éléments dans l'image non distordue doit rester un contour 2D droit. Ce travail propose une formalisation probabiliste originale continue de la présence d'alignement de primitives de contour ("edgels") au sein de l'image. Deux critères d'optimisation originaux ont été développés afin de maximiser cette probabilité, le premier basé sur un maximum de vraisemblance et le second sur l'entropie. Enfin, une méthode d'estimation de la probabilité qu'un pixel appartienne à un contour a été mise en place en se basant sur la propagation de l'erreur, supposée gaussienne, du capteur de la caméra. Des expériences réalisées, à la fois sur des déformations de synthèse et des caméras réelles à courte focale, comparent les performances de la méthode proposée en terme de précision de correction et de robustesse en présence de bruit par rapport aux techniques habituellement utilisées.

## Mots Clef

Étalonnage automatique de caméra, Contrainte de ligne droite, Distorsion radiale.

## Abstract

This article presents a simple and fully automatic radial distortion correction for uncalibrated short focal length cameras. The principle is to estimate the intrinsic parameters of a projection model from the analysis of a scene presenting straight three-dimensional elements. As introduced by [3, 10], the projection of the contours of these elements in the undistorted image must remain straight 2D edge. This work proposes a probabilistic and continuous formalization of the alignment of edge primitives ("edgels") in the image. Two original optimization criteria have been developed to maximize this probability, the first based on a maximum a posteriori and the second on entropy. Finally,

a method for estimating the probability that a pixel belongs to a edge has been established based on the error propagation, assumed Gaussian, of the camera sensor. Experiments have been realized, both on synthetic deformations and short focal length cameras to compare the actual performance of the proposed method in terms of accuracy of correction and robustness in the presence of noise against the usually used technics.

## Keywords

Camera self-calibration, Plumb-lines constraint, Radial distortion.

## 1 Introduction

L'utilisation de caméras à courte focale est de plus en plus courante dans les applications de vision par ordinateur. De par leur grand angle d'observation, ce type de caméra convient particulièrement aux applications de vidéo-surveillance ou de localisation. Sur ce type de système, la distorsion radiale due à l'objectif est si importante que la caméra ne peut plus être représentée par le modèle *pin-hole* seul. Afin d'obtenir une meilleure modélisation, deux techniques ont été explorées : la première consiste à garder le modèle *pin-hole* et y rajouter un jeu de paramètres modélisant une distorsion radiale dans le repère image [5]. La seconde consiste à changer de modèle de projection pour un plus adapté : le modèle unifié [4, 9]. Mais comme tout nouveau paramétrage, les deux solutions nécessitent un étalonnage. L'auto-étalonnage est une technique visant à retrouver les paramètres d'un modèle de caméra sans aucune intervention humaine. Il peut être effectué grâce à certaines propriétés de la scène observée. En effet, le monde actuel regorge de bâtiments et de structures aux profils rectilignes. Or, nous savons qu'un modèle de distorsion radial a pour propriété de déformer ces droites en courbes. L'idée sous-jacente est donc d'estimer les paramètres intrinsèques d'une caméra afin que celle-ci conserve les «droites droites» [3]. Nos travaux se sont concentrés sur l'estimation partielle des paramètres intrinsèques des caméras à courte focale dans le cadre de l'utilisation du modèle de projection unifié dans un environnement struc-

turé. Dans un premier temps, nous explorerons les différentes méthodes d'auto-étalonnage déjà existantes. Puis, nous présenterons notre méthode ainsi que les expérimentations réalisées. Enfin, nous conclurons sur les apports de cette nouvelle méthode et ses améliorations potentielles.

## 2 Travaux existants

Il existe de nombreuses méthodes d'estimation des paramètres intrinsèques d'un modèle de caméra. Une des plus simples consiste à acquérir une ou plusieurs images d'un environnement connu, ou d'un motif prédéfini comme une mire, puis à rechercher les paramètres optimaux en accord avec la scène ou le modèle théorique. Afin d'éviter l'utilisation de motif étalonné souvent coûteux, d'autres approches ont utilisé le mouvement de la caméra. Celles-ci consistent à détecter et à suivre des amers visuels (points, plans, lignes ...) le long d'une trajectoire connue ou inconnue [6, 5] et ainsi contraindre les paramètres intrinsèques. Cependant, il n'est pas toujours possible de présenter une mire à la caméra ou d'effectuer un déplacement du système d'acquisition. Parfois, l'accès même à la caméra n'est pas réalisable ou très fastidieux (microscopie, système embarqué). Pour résoudre ce problème, une série de méthodes ont été développées afin d'étalonner une caméra à partir d'une seule image. La majorité d'entre elles utilise la contrainte de ligne droite, car le monde regorge de structure rectiligne. Une des premières approches fut de détecter des courbes polygonales à partir des contours dans l'image puis à optimiser les paramètres intrinsèques afin que celles-ci soient les plus droites possible [3, 1]. Cette méthode nécessite la détection des contours puis l'extraction de segments et enfin leur regroupement en une courbe polygonale. Il arrive cependant qu'une droite soit occultée et forme deux segments, dans ce cas, cette méthode contraint de manière bien plus faible les paramètres intrinsèques, car elle ne détecte que deux entités plus petites. La méthode [10] pallie complètement à ce problème, car elle ne nécessite pas la détection de primitives au sein de l'image. Celle-ci consiste à créer un histogramme des orientations des contours et à minimiser l'entropie de cette distribution. En effet, une courbe est constituée d'une grande variété d'angles tandis qu'une droite n'en contient qu'un seul. Ainsi en minimisant l'entropie, le système converge dans un état où les contours tendent à être tous redressés. Cette méthode souffre pourtant d'un important défaut : la position de la droite dans l'image n'est pas prise en compte, car seules les orientations sont considérées. Cela conduit parfois à aligner des segments parallèles ensemble. De plus, cette méthode est discrète et le résultat dépend donc de la création de l'histogramme des orientations.

## 3 Principe de la méthode

Cette section expose le principe de la méthode de correction proposée. Si une caméra observe une scène totalement, ou partiellement, composée d'éléments présentant un contour droit, alors l'image non distordue formée

par la projection de cette scène comportera également des contours droits. Si la caméra utilisée engendre une distorsion radiale lors de la formation de l'image, la projection des contours sera courbe. A partir de cette constatation, nous proposons de définir une méthode de correction de la distorsion radiale. En utilisant un modèle de cette dernière, nous cherchons à estimer ses paramètres *de telle façon que le nombre de couples de points appartenant à la même droite de l'image soit maximal*. Nous définissons

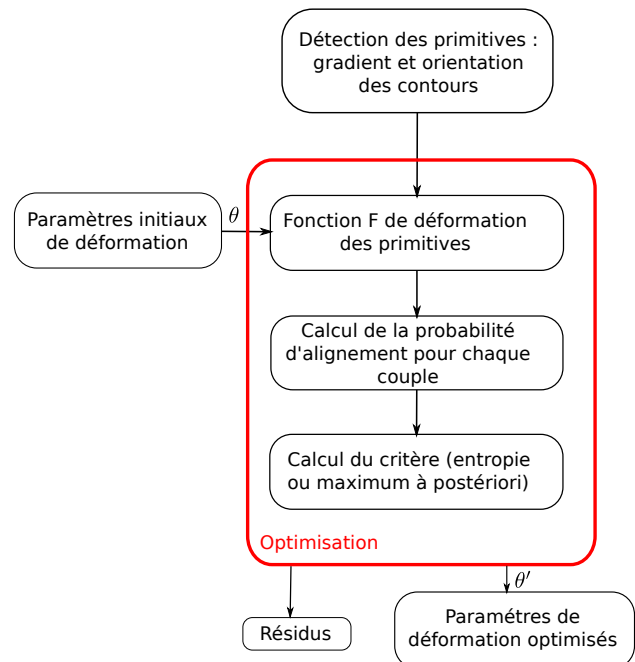


FIGURE 1 – Illustration de la méthode d'auto-étalonnage proposée.

par primitive de contour ("edgels") l'association, au sein d'une même entité, des coordonnées cartésiennes du point et de l'orientation locale du contour. Comme l'illustre la Figure 1, l'algorithme de correction automatique des distorsions est constitué des étapes suivantes :

1. Détecter des primitives :
  - Effectuer une détection probabiliste des contours
  - Effectuer un tirage par importance des points de contours détectés.
  - Calculer les gradients en  $x$  et  $y$  des points sélectionnés ( $g_x$  et  $g_y$ ).
  - Dédire la valeur de l'orientation des contours en utilisant la perpendiculaire au gradient :  $\arctan(g_x, g_y) - \frac{\pi}{2}$  ou notre méthode par correspondance de motifs.
  - Conserver les primitives de contours constituées des  $M$  points sélectionnés ( $x_i, y_i$ ) et de leurs orientations ( $\theta_i$ ).
2. Définir le vecteur initial  $\theta_0$  des paramètres de distorsion.
3. Maximiser la fonction de coût suivante :

- Appliquer la déformation  $F_\theta$  aux primitives.
- Estimer la probabilité d'alignement pour chaque couple de primitive.
- Combiner ces probabilités afin de calculer notre critère (le coût).

Chacune de ces étapes est maintenant détaillée.

### 3.1 Formalisation probabiliste

Soit une image  $I$  composée d'un ensemble  $\mathcal{P}$  de  $N$  pixels  $\{P_i\}_{i \in [1, N]}$ . On associe à chaque point de l'image  $P_i$  une variable aléatoire  $c_i$  qui suit une loi de Bernoulli et prend la valeur 1 lorsque le pixel  $P_i$  est un point de contour (voir la section 3.5). On note alors  $Q_i$  la primitive associée au point  $P_i$  et qui contient  $x_i, y_i$  (coordonnées) et  $\theta_i$  (tangente aux contours). On forme alors l'ensemble  $\mathcal{C}$  des couples de primitives possibles  $\{C_{ij}\}_{(i,j) \in [1, N]^2, i \neq j}$ . Pour chaque couple  $C_{ij}$ , on associe une variable aléatoire  $a_{ij}$  qui suit une loi de Bernoulli et prend la valeur 1 lorsque les primitives  $Q_i$  et  $Q_j$  sont alignées (voir la section 3.6). De ce fait, deux points de l'image sont sur la même droite si : 1) ce sont des points de contour, et 2) ils possèdent des primitives alignées. En considérant les  $c_i$  et  $a_{ij}$  indépendantes, la probabilité que deux points  $P_i$  et  $P_j$  appartiennent à la même droite dans l'image est donnée par :

$$p(c_i, c_j, a_{ij} | \mathbf{z}) = p(c_i | \mathbf{z}) p(c_j | \mathbf{z}) p(a_{ij} | \mathbf{z}) \quad (1)$$

où  $\mathbf{z}$  représente l'observation faite dans l'image. En appliquant la règle de Bayes, cette distribution *a posteriori* se calcule :

$$p(c_i, c_j, a_{ij} | \mathbf{z}) = C^{-1} p(\mathbf{z} | c_i) p(\mathbf{z} | c_j) p(\mathbf{z} | a_{ij}) p(c_i, c_j, a_{ij}) \quad (2)$$

$C^{-1}$  est un terme de normalisation et on considère des distributions *a priori* équiprobables :

$$p(c_i, c_j, a_{ij} | \mathbf{z}) \propto p(\mathbf{z} | c_i) p(\mathbf{z} | c_j) p(\mathbf{z} | a_{ij}) \quad (3)$$

### 3.2 Critères d'optimisation

Dans cette partie, nous définissons deux critères permettant de quantifier le taux de contours droits dans l'image. Le premier critère consiste à maximiser la vraisemblance associée au nombre de couples de points de l'image appartenant à la même droite :

$$C_1 \doteq \sum_{(i,j) \in [1, N]^2, i \neq j} p(\mathbf{z} | c_i) p(\mathbf{z} | c_j) p(\mathbf{z} | a_{ij}) \quad (4)$$

Une autre approche consiste à utiliser un raisonnement similaire à celui de [10]. La probabilité calculée en (3) traduit une quantité d'information associée à l'événement « le couple de points appartient à la même droite dans l'image ». L'entropie associée au couple de points  $P_i, P_j$  se calcule comme suit :

$$H_{i,j} = -[p(\mathbf{z} | c_i, c_j, a_{ij}) \log(p(\mathbf{z} | c_i, c_j, a_{ij})) + (1 - p(\mathbf{z} | c_i, c_j, a_{ij})) \log(1 - p(\mathbf{z} | c_i, c_j, a_{ij}))] \quad (5)$$

Il est alors possible de définir un second critère basé sur un minimum d'entropie associée à l'ensemble  $\mathcal{C}$  des couples de points en additionnant leurs entropies (sur une hypothèse d'indépendance des couples) :

$$C_2 \doteq - \sum_{(i,j) \in [1, N]^2, i \neq j} H_{ij} \quad (6)$$

### 3.3 Estimation des paramètres d'une fonction de correction de la distorsion

Soit  $P_i^* = F(P_i, \theta)$ , une fonction de correction de distorsion paramétrique (de vecteur de paramètres  $\theta$ ) qui associe, au pixel  $P_i$  de l'image de départ, un pixel  $P_i^*$  corrigé. Cette fonction modifie l'orientation du contour en chaque point, (on notera donc  $Q_i^*$  la primitive corrigée) mais ne joue pas sur la probabilité qu'un point soit un point de contour. Aussi, en définissant une variable aléatoire  $a_{ij}^*$  qui suit une loi de Bernoulli et prend la valeur 1 lorsque les primitives  $Q_i^*$  et  $Q_j^*$  associées aux points  $P_i^*$  et  $P_j^*$  de l'image corrigée sont alignées, il est possible de définir deux estimations de  $\theta$  à partir des deux critères  $C_1$  et  $C_2$  par :

$$\hat{\theta}_1 = \arg \max_{\theta} \sum_{(i,j) \in [1, N]^2, i \neq j} p(\mathbf{z} | c_i) p(\mathbf{z} | c_j) p(\mathbf{z} | a_{ij}^*) \quad (7)$$

avec  $p(\mathbf{z} | a_{ij}^*)$  la probabilité que  $Q_i^* = F(Q_i, \theta)$  et  $Q_j^* = F(Q_j, \theta)$  soit alignées.

$$\hat{\theta}_2 = \arg \max_{\theta} \sum_{(i,j) \in [1, N]^2, i \neq j} H_{ij}^* \quad (8)$$

où  $H_{ij}^*$  est l'entropie associée aux points  $P_i^*$  et  $P_j^*$  de l'image corrigée. L'estimation peut alors être réalisée par une méthode d'optimisation non-linéaire.

### 3.4 Approximation par sous-échantillonnage préférentiel

La méthode proposée ci-dessus est très lourde à mettre en oeuvre en terme de ressources et de temps de calcul pour des images de tailles classiques. A titre d'exemple, sur une image de taille  $640 \times 480$ , il y a  $[(640 \times 480)^2]$  couples de points possible, ce qui rend le traitement impossible. Nous proposons donc une stratégie de sous-échantillonnage préférentiel (aussi nommé : tirage par importance) de cet ensemble. On considère l'ensemble des pixels d'une image comme une fonction de masse dont la probabilité d'appartenir à un contour, associée à chaque pixel, est proportionnelle à  $p(\mathbf{z} | c_i)$ . L'objectif de l'échantillonnage préférentiel (utilisé comme sous-échantillonnage dans le cas présent) est d'approcher la fonction de masse par une méthode de Monte-Carlo. La figure 2 illustre le sous-échantillonnage préférentiel. On forme un ensemble de couples  $\{P_i, p(c_i)\}_{P_i \in \mathcal{P}}$ , représentatifs d'un ensemble de vraisemblances de points appartenant à des contours. On cherche à approximer cet ensemble par un tirage de  $K \ll N$  échantillons où chaque échantillon possède la même vraisemblance. Pour cela, on effectue un tirage avec remise, selon la loi  $p(\mathbf{z} | c_i)$ . Ainsi, un point possédant une

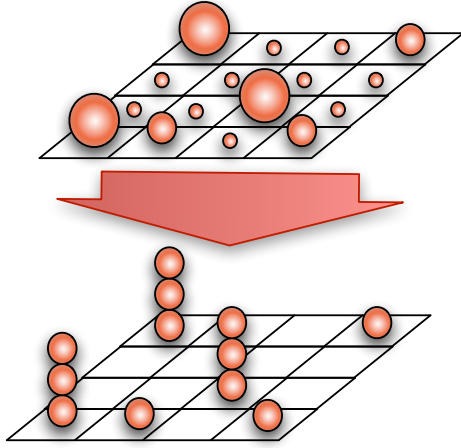


FIGURE 2 – Illustration du sous-échantillonnage par tirage d'importance. Le haut de la figure montre une image où chaque boule représente la probabilité  $p(\mathbf{z}|c_i)$  que le pixel associé soit un contour. Le volume de la boule est proportionnel à  $p(\mathbf{z}|c_i)$ . Le bas de la figure montre le résultat d'un sous-échantillonnage par tirage d'importance. Les pixels de forte probabilité sont tirés plusieurs fois tandis que les pixels de faible probabilité ne sont pas tirés.

forte probabilité  $p(\mathbf{z}|c_i)$  sera tiré plusieurs fois et un point possédant une faible probabilité aura peu de chance d'être tiré. Soit  $\tilde{\mathcal{P}}\{\tilde{P}_i\}_{i:1,\dots,K}$  l'ensemble des points issus de l'échantillonnage d'importance. Cet ensemble permet de faire l'approximation suivante :

$$\sum_{(i,j) \in [1,N]^2, i \neq j} p(\mathbf{z}|c_i)p(\mathbf{z}|c_j)p(\mathbf{z}|a_{ij}^*) \approx \sum_{(i,j) \in [1,K]^2, \tilde{P}_i \neq \tilde{P}_j} p(\mathbf{z}|\tilde{a}_{ij}^*) \quad (9)$$

où  $\tilde{a}_{ij}^*$  est une variable aléatoire qui suit une loi de bernouilli et prend la valeur 1 lorsque les points  $\tilde{P}_i^*$  et  $\tilde{P}_j^*$  de l'image corrigée par le modèle de paramètres  $\theta$  possèdent des tangentes (en terme de contour) alignées. Les critères  $C_1$  et  $C_2$  sont alors approximés par :

$$C_1 \approx \sum_{(i,j) \in [1,K]^2, \tilde{P}_i \neq \tilde{P}_j} p(\mathbf{z}|\tilde{a}_{ij}^*) \quad (10)$$

$$C_2 \approx \sum_{(i,j) \in [1,K]^2, \tilde{P}_i \neq \tilde{P}_j} p(\mathbf{z}|\tilde{a}_{ij}^*) \log(p(\mathbf{z}|\tilde{a}_{ij}^*)) + (1 - p(\mathbf{z}|\tilde{a}_{ij}^*)) \log(1 - p(\mathbf{z}|\tilde{a}_{ij}^*)) \quad (11)$$

### 3.5 Calcul de la vraisemblance $p(\mathbf{z}|c_i)$

Les détecteurs de contours les plus communs (Canny [2], Sobel, zéros-crossing [7],...) commence par appliquer un filtre de convolution afin d'obtenir une intensité en chaque pixel. Puis cette intensité est ensuite interprété de manière

à fournir un resultat binaire. Ceux-ci ne prennent donc pas en compte la quantité de bruit contenu dans l'image et, de ce fait, ils ne propagent pas l'incertitude de détection sur leurs résultats : Il ne sont donc pas probabiliste. Le filtre de Canny comme les autres détecteurs fournit de très bon résultat mais sa performance se dégrade fortement lorsque le bruit augmente. Afin d'être opérationnel en toute condition et d'être en accord avec notre théorie, il nous faut calculer la probabilité qu'un point de l'image soit un contour au vue du bruit du capteur ainsi que du détecteur choisi. Ce problème a déjà été mis en lumière par [7] qui a réussi à propager l'incertitude au travers du détecteur de contour zéros-crossing. Malheureusement, cette méthode fonctionne grâce à l'expression linéaire du laplacien discret et à de lourdes hypothèses. Nous proposons, sous l'hypothèse que le bruit est gaussien et que la corrélation inter-pixélique est nulle, d'estimer la probabilité des résultats d'un détecteur de contour quelconque grâce à la méthode de Monte-Carlo. Pour cela, nous bruitons l'image d'origine avec un bruit gaussien, et effectuons la détection de contour. En répétant ce processus, la somme des résultats binaires du détecteur divisé par le nombre d'itérations converge vers la carte de probabilité des contours.  $p(\mathbf{z}|c_i)$  est alors égale à la valeur de la  $i$ -eme case de cette carte. Cette méthode simple permet de transformer n'importe quel détecteur de contours fournissant un résultat binaire en un détecteur probabiliste.

### 3.6 Calcul de la vraisemblance $p(\mathbf{z}|\tilde{a}_{ij}^*)$

**Vraisemblance d'alignement de deux primitives.** Soit  $Q_i$  et  $Q_j$  un couple de primitives défini par les coordonnées des points associés et les tangentes locales aux contours.  $\vec{V}_{ij}$  est le vecteur reliant leur position respective (voir Figure 3). Notons  $\psi_{ij}$  l'angle formé entre l'abscisse et le vecteur  $\vec{V}_{ij}$  ainsi que  $\phi_i$  et  $\phi_j$  ceux formés par les directions locales aux contours ( $\vec{T}_i$  et  $\vec{T}_j$ ) avec l'abscisse. Nous constatons que si  $\psi_{ij} = (\phi_i \bmod \pi)$  et  $\psi_{ij} = (\phi_j \bmod \pi)$  alors les primitives sont alignées. Afin de définir une vrai-

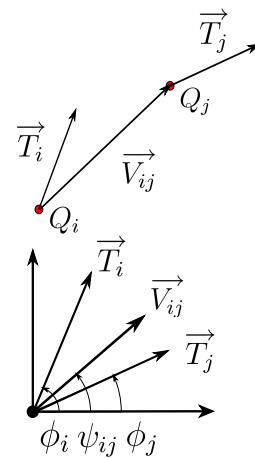


FIGURE 3 – Illustration du critère de colinéarité proposé

semblance de colinéarité, nous posons l'hypothèse que le bruit de détection de l'angle des tangentes suit une loi normale et que l'incertitude sur la position du point est négligeable. Alors, la vraisemblance de colinéarité est d'autant plus élevée que  $\phi_i$  et  $\phi_j$  sont proches de  $\psi_{ij}$ . Ce qui se formalise par :

$$p(\mathbf{z}|\tilde{a}_{ij}^*) = \frac{1}{\sigma_i \sigma_j 2\pi} \exp\left(-\frac{(\phi_i - \psi_{ij})^2}{2\sigma_i^2}\right) \exp\left(-\frac{(\phi_j - \psi_{ij})^2}{2\sigma_j^2}\right) \quad (12)$$

où  $\sigma_i$ ,  $\sigma_j$  représentent les incertitudes angulaires respectives des primitives  $Q_i$  et  $Q_j$ . Nous avons tout d'abord utilisé la perpendiculaire au gradient local comme estimation de la direction de la tangente. Puis, nous avons développé une méthode par correspondance de motifs afin d'améliorer la précision et la robustesse au bruit.

**Présentation rapide du modèle de caméra unifié.** Le modèle de caméra unifié a été initialement proposé par [4] pour des capteurs catadioptriques centraux. Cependant, sa modélisation s'accorde avec la représentation des caméras à courte focale. La correction d'un point 2D  $P$  dans l'image en un point  $p$  s'effectue en deux étapes successives :

1. une projection du point  $P$  sur la sphère unitaire de centre  $X_c = u_0, Y_c = v_0, Z_c = \xi$  en un point  $Q$ ,
2. une reprojection perspective de  $Q$  sur le plan normalisé  $Z_c = 1$  en un point  $p$ ,

Le vecteur d'optimisation  $\theta$  contient donc la position du centre optique dans l'image  $u_0$  et  $v_0$  ainsi que  $\xi$ .

#### Déformation des primitives par la distorsion radiale.

Lors de l'optimisation, l'algorithme modifie les paramètres du modèle de caméra unifié  $\theta$  de façon à minimiser nos critères (7) et (8). A chaque nouvelle évaluation, il est nécessaire de mettre à jour les primitives avec ces mêmes paramètres, mais nous n'appliquons en aucun cas ce changement sur l'image d'origine. En effet, cela prendrait un temps considérable, nécessiterait l'interpolation de l'image corrigée et la re-détection des contours. Les primitives de contours ne sont calculées qu'une seule fois et nous appliquons la distorsion directement sur ces dernières. Puisque chacune d'entre elles est constituée des coordonnées cartésiennes d'un point noté  $(x_i, y_i)$  et d'un angle  $\phi_i$ , il est nécessaire de mettre à jour les coordonnées du point, mais aussi l'angle de la primitive. Pour cela nous introduisons un nouveau point de coordonnées :  $(x_i + dl \cos(\phi_i), y_i + dl \sin(\phi_i))$  placé dans la direction de la tangente et infiniment proche du point d'origine,  $dl$  représente une très petite valeur positive. En appliquant la distorsion à ces deux points, nous obtenons la position de la nouvelle primitive  $(x'_i, y'_i)$  et l'angle du nouveau vecteur tangent  $\phi'_i$  comme sur la figure 4. Cette solution calculatoire a pour but d'éviter le calcul de la jacobienne de la transformation [10]. En effet, le modèle unifié étant fortement non linéaire, ce calcul s'avère complexe et débouche sur une expression à fort coût calculatoire.

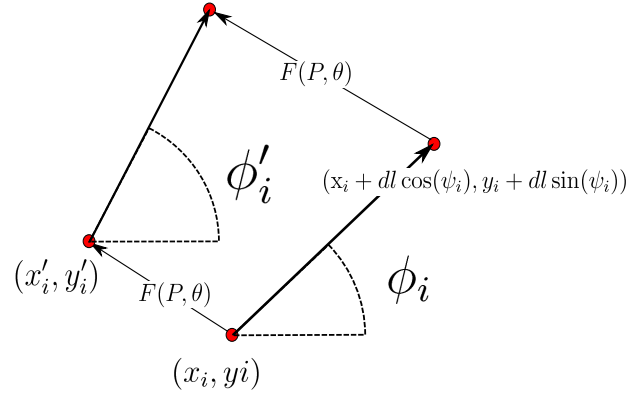


FIGURE 4 – Illustration de la méthode d'application de la distorsion radiale sur les primitives

#### Détection de la direction d'un contour par correspondance de motif.

Lors de nos tests, nous avons constaté que la tangente au contour, calculée uniquement avec la direction du gradient, était extrêmement sensible au bruit. Afin de réduire la variance de la détection et donc d'augmenter la qualité du résultat, nous avons utilisé la méthode des moindres carrés entre un motif généré à partir d'un modèle mathématique de contour et des pixels du voisinage du point sélectionné. Pour cela nous avons choisi un modèle de contour 2D très simple que nous avons re-paramétré dans le repère polaire :

$$e(x, y) = I_g + \frac{I_d - I_g}{2} \operatorname{erf}\left(\frac{x \cos(\phi) - y \sin(\phi) + \rho}{\sqrt{2}\sigma}\right) \quad (13)$$

où  $\phi$  représente l'angle du contour,  $\rho$  une translation permettant d'ajuster au mieux le contour,  $\sigma$  sa largeur,  $I_d$  et  $I_g$  l'intensité maximale à gauche et à droite du contour ainsi généré.

Ces paramètres sont retrouvés grâce à l'algorithme de Levenberg-Maquardt auquel on fournit la différence entre le motif (classiquement d'une taille de  $15px \times 15px$ ) et l'image. Afin que l'optimisation se fasse au plus vite, nous n'utilisons cette méthode que sur les points résultant de l'échantillonnage par importance et initialisons les paramètres avec les valeurs données par la direction et l'intensité du gradient. La figure 5 illustre la qualité du résultat.

## 4 Expérimentations

Cette partie présente les expérimentations menées pour évaluer quantitativement et qualitativement, la précision de la correction ainsi que sa robustesse au bruit devant différents types d'images plus ou moins complexes. Dans toutes nos expériences, les images sont en niveaux de gris et les niveaux d'intensités vont de 0 à 255. Les valeurs des écarts-types sont données dans ces mêmes unités. Lorsque le détecteur de contour probabiliste est utilisé, le détecteur binaire sous-jacent est Canny [2].



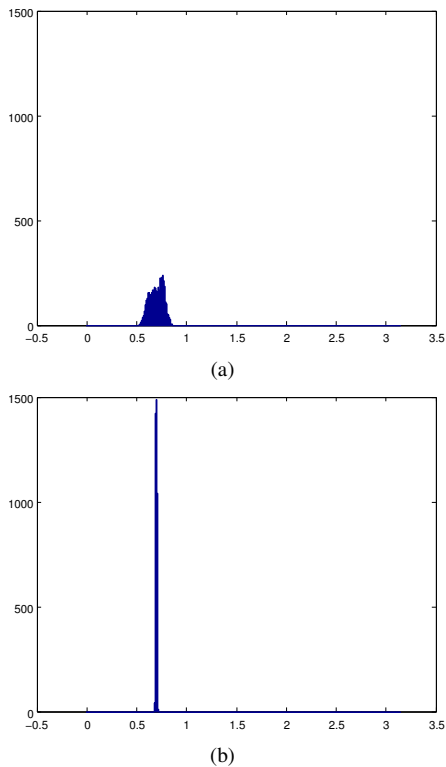


FIGURE 5 – Histogrammes des orientations des détections sur une image test (constitué de bandes diagonales parallèles noires sur fond blanc) et additionnée à un bruit gaussien ( $\sigma = 5$ ) de moyenne nulle. (a) Perpendiculaire au gradient (écart-type : 0.0693 radians, 4.00 degrés). (b) Méthode par correspondance de motif (écart-type : 0.0080 radians, 0.45 degrés)

## 4.1 Méthodologie

**Données synthétiques.** Dans un premier temps, nous avons travaillé sur des données synthétiques. Pour cela nous avons généré un motif (Figure 6) contenant des droites parfaites auquel nous avons rajouté un bruit gaussien d'écart-type  $\sigma_b$  a la valeur de chaque pixel. Ensuite, nous effectuons la détection des contours, puis des primitives, et ensuite nous appliquons une distorsion radiale arbitraire ( $\theta_t$ ) à l'aide du modèle unifié. Le but est de retrouver les paramètres de distorsions grâce à notre méthode et d'estimer la qualité du résultat. Afin d'obtenir des critères quantitatifs sur la correction de la distorsion, nous présentons les résultats d'étalonnage ainsi que le score RMS (en pixel) calculé entre la position des primitives sur l'image de synthèse et sur celle corrigée par les paramètres de distorsions optimisés. En faisant varier la valeur du bruit gaussien ainsi que la distorsion de l'image, nous avons mis à l'épreuve notre critère. Afin de valider aussi le détecteur de contour probabiliste et la correspondance de motif, nous avons répété cette expérience avec les méthodes classiques.

**Données réelles.** Afin de valider les améliorations de notre méthode sur des caméras réelles, nous avons fait l'ac-

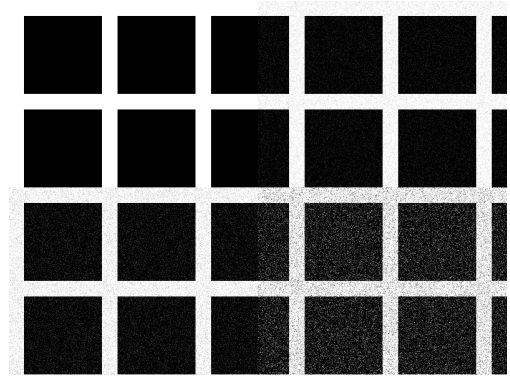


FIGURE 6 – Figure synthétique utilisée, elle est constituée de droite rectiligne de largeur 25px et espacée de 125px. La taille totale de l'image est de  $600 \times 800px$ . Les quatre quarts de l'image ont été bruités avec des valeurs différentes de  $\sigma$  : 0 et 10 en haut puis 30 et 70 en bas

quisition d'images avec une caméra Pixelink de résolution  $720 \times 480px$  et muni d'un objectif a courte focale (3mm). Diverses scènes ont été observées, contenant plus ou moins de droites (présenté à la figure 8). Dans le but de comparer la méthode à l'existant, nous avons choisi de calibrer cette caméra avec la Toolbox Matlab écrite par Christopher Mei [8]. Nous la prenons comme vérité terrain, car celle-ci permet de trouver avec précision la calibration d'une caméra selon le modèle unifié grâce à une mire damier. Nous avons aussi ré-implémenter le critère basé sur l'entropie de Hough (entropie de l'histogramme des orientations) utilisé dans [10] dans le but d'analyser les apports de notre nouveau critère. Afin de comparer uniquement les critères entre eux, nous avons utilisé les mêmes primitives pour les deux méthodes.

La pertinence est estimée par le score RMS entre les primitives corrigées par l'étalonnage de la Toolbox et les mêmes primitives corrigées par l'une des deux méthodes. Les résultats sont consignés dans le tableau 2.

**Implémentation et complexité algorithmique.** L'algorithme proposé a une complexité en  $O(n^2)$ , où  $n$  représente le nombre de primitives après l'échantillonnage par importance et a été implémenté en Matlab. Pour donner un ordre d'idée du temps d'exécution, le traitement d'une image est d'environ 20 minutes sur un *Core 2 Duo 2.60Ghz* en utilisant 1000 primitives. En détail les temps se répartissent comme suit :

1. Détecter les contours
  - Canny : 0.2s
  - Détecteur probabiliste : 2 minutes
2. Extraire les primitives
  - Perpendiculaire au gradient : 0.1s
  - Correspondance de motif : 12 minutes
3. Optimiser : 6 minutes

Les améliorations apportées par 1) le détecteur probabiliste et 2) la correspondance de motifs sont très coûteuses et une

version basique sans ses deux améliorations s'exécute en environ 6 minutes. Tous ces temps sont donnés à titre indicatif et les algorithmes implantés n'ont fait l'objet d'aucune optimisation.

## 4.2 Résultats et analyse

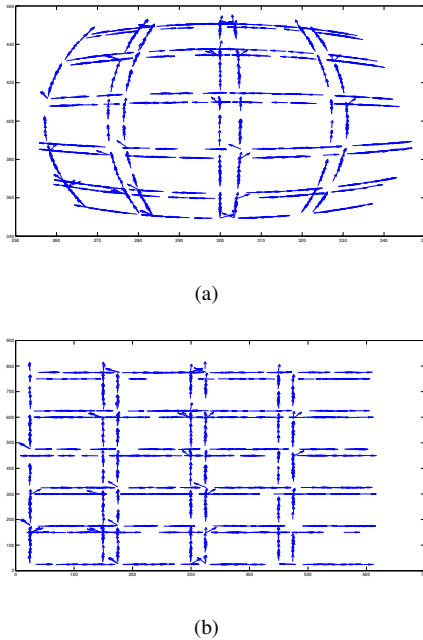


FIGURE 7 – (a) : Primitives détectées sur l'image synthétique après une distorsion par le modèle unifié ( $u_0 = 300$ ,  $v_0 = 400$ ,  $\gamma = 300$ ,  $\xi = 0.4$ ). C'est une déformation typique des caméras à courte focale et des *fisheyes*. (b) : Primitives corrigées après optimisation du coût.

	$RMS(a)$	$RMS(b)$	$RMS(c)$
$\sigma = 0$	0.15	0.1491	<b>0.125</b>
$\sigma = 10$	0.9813	<b>0.515</b>	1.6869
$\sigma = 20$	2.9984	<b>0.9570</b>	5.3413
$\sigma = 30$	<b>2.7633</b>	D	D
$\sigma = 70$	<b>10.6232</b>	D	D
$\sigma > 70$	D	D	D

TABLE 1 – Résistance au bruit de différentes méthodes, D signifie que l'algorithme diverge. (a) Scores RMS de la méthode proposée. (b) Scores RMS en utilisant la perpendiculaire au gradient à la place de la correspondance de motif. (c) Scores RMS de en utilisant Canny binaire à la place du détecteur probabiliste.

Le tableau 1 montre que la méthode comprenant la correspondance de motif et la détection probabiliste est capable de corriger une image très distordue (Fig.7) et est extrêmement robuste au bruit. Cependant, dans la réalité, il est rare que le bruit pixélique dépasse un écart-type de 5, mais le but premier était de valider l'apport de chacune

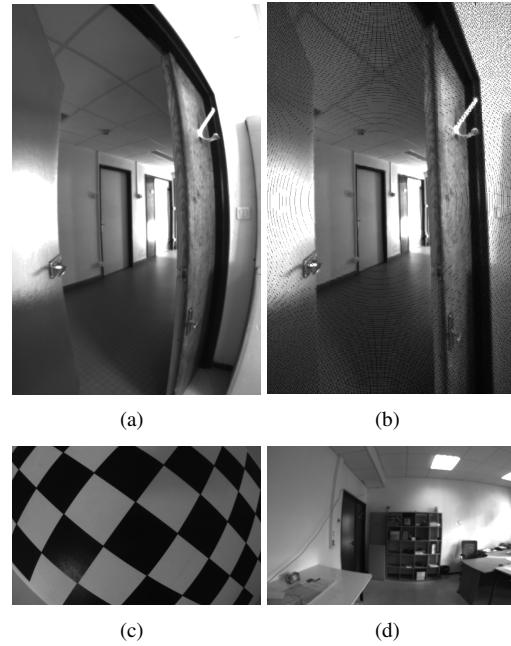


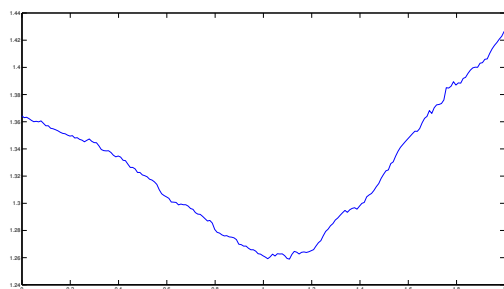
FIGURE 8 – (a) : Image réelle distordue (Couloir). (b) : Image corrigée par notre algorithme. (c) Image d'une mire damier. (d) Image d'un bureau.

	$RMS(a)$	$RMS(b)$
Mire	1.14	9.9925
Bureau	5.339	9.9917
Couloir	8.26	15.175*

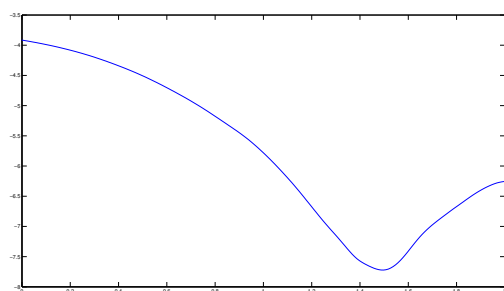
TABLE 2 – Comparaison entre les résultats des deux méthodes sur les trois images de test. (a) Utilisation de notre critère d'entropie. (b) Utilisation de l'entropie de l'histogramme des orientations selon [10]. (\*) La méthode (b) converge vers un minimum local pour l'image du couloir.

des contributions proposées. Lors de l'utilisation de la perpendiculaire au gradient à la place de la correspondance de motif, nous avons remarqué que l'algorithme se comporte correctement pour un bruit raisonnable. Un flou gaussien ( $\sigma = 2px$ ) a été appliqué avant le calcul des orientations. Ce dernier a effacé le bruit et a contribué à ce résultat. Sans ce flou, la méthode ne fonctionne qu'avec les données parfaites. Il est aussi intéressant de constater que pour un bruit assez faible, l'utilisation du détecteur de Canny (réponse binaire) fournit un résultat identique au détecteur probabiliste. Lorsque le bruit augmente, le tirage par importance sélectionne de plus en plus de primitives sur des détections aberrantes tandis que sur la carte de probabilité, l'importance des contours est pondérée et les fausses détections ont donc moins de chance d'être tirées. Enfin, le tableau 2 recense les scores RMS de la méthode utilisant l'entropie de Hough, la méthode de référence, en comparaison aux critères proposés. Sur toutes les images tests, le critère proposé fournit des résultats de meilleure qualité. La figure 9 montre l'évolution du critère de la méthode de référence





(a)



(b)

FIGURE 9 – Tracé de la fonction de cout pour les deux méthodes avec l'image du couloir en fonction de  $\xi$  (paramètre principal du modèle unifié). (a) Entropie de l'histogramme des orientations (b) Notre critère.

et de la méthode proposée, en fonction de la valeur du paramètre de distorsion, sur l'image du couloir. La méthode de l'entropie de Hough converge vers un minimum local. L'entropie de l'histogramme des orientations contient clairement un grand nombre de minimaux locaux, car celle-ci se base sur une discrétisation et ne prend pas en compte la position de la droite dans l'image. Notre critère, quant à lui, est continu et respecte la position de la droite au sein de l'image, il ne possède que très rarement de tel minimum.

## 5 Conclusion et perspectives

Nous avons présenté dans cet article une méthode d'étalonnage automatique de la distorsion radiale dans le cadre du modèle de caméra unifié. Cette méthode passe par la détection de contour, puis l'extraction de primitives ("edges") constituées d'une position et d'une orientation. Ensuite, nous avons défini un nouveau critère continu permettant de quantifier de manière probabiliste la présence d'alignements au sein des primitives. Enfin, nous avons mis en évidence deux méthodes d'évaluation du coût : un maximum de vraisemblance et une minimisation d'entropie.

Les expérimentations ont montré que cette méthode est particulièrement robuste au bruit, d'une part grâce à l'approche probabiliste et à la propagation des incertitudes tout au long du processus, mais aussi parce qu'il n'y a pas de détection d'amer au sein de l'image. En effet l'utilisation de

primitives locales offre la possibilité d'utiliser la contrainte de ligne droite même si celle-ci ne sont pas continues. Le critère proposé présente un bassin de convergence particulièrement régulier et comportant peu de minima locaux, contrairement à d'autres critères existants.

Cependant, dans ces travaux, une hypothèse d'environnement complètement structuré a été réalisée, la non-structure étant vue comme du bruit de mesure. La méthode fonctionne tout de même si la scène contient quelques courbes, mais celles-ci pourraient être à l'origine d'un biais. Afin que l'auto-étalonnage puisse travailler sur des scènes comprenant un nombre important de courbes, il serait judicieux d'insérer une nouvelle probabilité :  $p(d_i)$ , représentant la probabilité que le point  $i$  soit sur une droite ou non. Une des difficultés d'estimation de  $p(d_i)$  provient du fait que l'image d'origine est distordue, donc toutes les droites sont courbes.

## Remerciements

Ces travaux ont été financés par la région Auvergne et l'Europe

## Références

- [1] Duane C. Brown. Close-range camera calibration. *Photogrammetric engineering*, 37(8) :855–866, 1971.
- [2] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6) :679–698, 1986.
- [3] F. Devernay and O. Faugeras. Straight lines have to be straight. In *In SPIE, volume 2567*, 2001.
- [4] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical implications. *Computer Vision ECCV 2000*, pages 445–461, 2000.
- [5] Jean M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration? In *ECCV '98*, pages 158–174, London, UK, 1998. Springer-Verlag.
- [6] Jean M. Lavest, M. Viala, and M. Dhome. Quelle précision pour une mire d'étalonnage. *Traitement du Signal*, 16(3) :241–254, 1999.
- [7] David H. Marimont and Y. Rubner. A probabilistic framework for edge detection and scale selection. In *in 'Proc. 6th Int. Conf. on Computer Vision*, pages 207–214, 1998.
- [8] C. Mei and P. Rives. Calibrage non biaise d'un capteur central catadioptrique. In *RFIA*, January 2006.
- [9] Y. Mezouar, J. Courbon, and P. Martinet. Evaluation of the unified model on the sphere for fisheye cameras in robotic applications. *Advanced Robotics*, To appear.
- [10] E. Rosten and R. Loveland. Camera distortion self-calibration using the plumb-line constraint and minimal hough entropy. *Machine Vision and Applications*, October 2009.