

Comparaison de performances des détecteurs KAZE, ORB, SURF

Ratsimbazafy T.H.¹, Randriamitantsoa P.A.²

Laboratoire de recherche Télécommunication, Automatique, Signal Et Images (LR-T.A.S.I)

Ecole Doctorale en Sciences et Techniques de l'Ingénierie et de l'Innovation (ED -STII)

Équipe d'Accueil Doctorale Télécommunication, Automatique, Signal et Images (EAD-TASI)

BP 1500, Ankatso- Antananarivo 101 –Madagascar

¹tsiorha@yahoo.fr, ²rpauguste@gmail.com

Résumé

L'image est largement utilisée dans de nombreuses applications basées sur la vision.

Le moment et la précision de l'enregistrement d'images basé sur des caractéristiques dépend principalement de l'efficacité de calcul et la robustesse du descripteur de la caractéristique sélectionnée.

Cet article présente une comparaison des algorithmes, KAZE, ORB, SURF. Il élucide également un dilemme critique : quel algorithme est le plus invariant à l'échelle, à la rotation et aux changements de point de vue ainsi qu'à diverses attaques

Mots clés: KAZE, ORB, SURF, détecteurs, images

Abstract

The image is widely used in many vision-based applications.

The timing and accuracy of feature-based image registration depends mainly on the computational efficiency and robustness of the descriptor of the selected feature.

This paper presents a comparison of the algorithms, KAZE, ORB and SURF. It also elucidates a critical dilemma: which algorithm is the most invariant to scale, rotation and point of view changes as well as to various attacks

Keywords: KAZE, ORB, SURF, detectors, images

1. Introduction

Il existe de nombreux algorithmes dédiés à l'extraction de caractéristiques des images. Beaucoup d'entre eux fonctionnent de la même manière qu'un spiropgraphe. C'est à dire fait le tour en se cognant contre les bords jusqu'à ce qu'il couvre, avec un peu de chance, chaque point de l'image. Ainsi, un algorithme se déplace autour d'une image et en extrait des informations intéressantes. Ce processus est appelé détection de caractéristiques.

1.1 KAZE

Le terme KAZE vient du mot japonais qui signifie « vent ». P. F. Alcantarilla et al. ont proposé en 2012 des caractéristiques KAZE qui exploitent l'espace d'échelle non linéaire par un filtrage de diffusion non linéaire. Cela permet de rendre le flou dans les images localement adaptable aux points caractéristiques, réduisant ainsi le bruit et conservant simultanément les limites des régions dans les images du sujet.

Le détecteur KAZE est basé sur le déterminant normalisé de la matrice de Hesse qui est calculé à plusieurs niveaux d'échelle. Les maxima de réponse du détecteur sont captés sous forme de points caractéristiques à l'aide d'une fenêtre mobile. La description des caractéristiques introduit la propriété

d'invariance de rotation en trouvant l'orientation dominante dans un voisinage circulaire autour de chaque caractéristique détectée.

L'équation ci-contre montre la formule de diffusion non linéaire standard :

$$\frac{\partial X}{\partial t} = \text{div} (c(x, y, t) \cdot \nabla X) \quad (01)$$

Où c est la fonction de conductivité, div est la divergence, ∇ est l'opérateur de gradient et X est la luminance de l'image

L'expression de la conductivité est :

$$c(x, y, t) = g(|\nabla X_\sigma(x, y, t)|) \quad (02)$$

où la fonction ∇X_σ est le gradient d'une version gaussienne lissée de l'image originale X avec un écart type σ . La fonction de conductivité g est choisie pour promouvoir de vastes régions,

$$g = \frac{1}{1 + \frac{|\nabla X_\sigma|^2}{k^2}} \quad (03)$$

où la constante k est le facteur de contraste qui contrôle le niveau de diffusion. La version discrétisée de l'équation (1) peut être exprimée en notation matrice-vecteur comme

$$\frac{X^{i+1} - X^i}{\tau} = \sum_{l=1}^m A_l(X^i)X^{i+1} \quad (04)$$

où X^i et X^{i+1} sont les images lissées respectivement au niveau actuel et au niveau suivant, m est le nombre des dimensions de l'image X (deux dans notre cas), τ est la taille du pas de temps, et A_l est la matrice qui encode les diffusions pour chaque dimension de l'image (dérivées le long du l -ème axe de coordonnées).

Pour construire l'espace d'échelle, KAZE lisse l'image originale en utilisant un noyau gaussien d'un écart type σ^0 pour réduire le bruit. Ensuite, le paramètre k est obtenu comme le 70e percentile de l'histogramme de gradient de cette version lissée de l'image. Afin de calculer X^{i+1} à partir X^i dans l'équation (4), il faut résoudre un système linéaire tridiagonal qui peut être réalisé en utilisant l'algorithme de Thomas.

Une fois l'espace d'échelle construit, il est nécessaire de détecter les points clés, qui sont des points avec certaines caractéristiques (indépendance de la position, robustesse contre les transformations de l'image, et échelle indépendance).

Un pixel sera considéré comme un point clé s'il est le maximum de son voisinage lorsque la matrice de la Hesse est calculée. Le

déterminant de la matrice de Hesse aux différents niveaux de l'échelle σ^2 est calculé comme suit

$$X_{\text{Hessian}}^i = \sigma^2(X_{xx}X_{yy} - X_{xy}^2) \quad (05)$$

où X_{xx} et X_{yy} sont les dérivés horizontaux et verticaux de second ordre, respectivement, et X_{xy} est le dérivé croisé de second ordre.

Cet ensemble de first et de dérivés de second ordre est approximé par 3×3 filtre de Scharr qui offrent une meilleure invariance de rotation que les autres filtres populaires

1.2 ORB

ORB (Oriented FAST and Rotated BRIEF) a été développé dans les laboratoires OpenCV par Ethan Rublee, Vincent Rabaud, Kurt Konolige, et Gary R. Bradski en 2011, comme une alternative efficace et viable au SIFT (Scale Invariant Feature Transform) et au SURF (Speeded-Up Robust Features).

ORB a été conçue principalement parce que SIFT et SURF sont des algorithmes brevetés.

L'ORB s'appuie sur le célèbre détecteur de points clés FAST (Features from Accelerated Segments Test) et le descripteur BRIEF (Binary Robust Independent Elementary

Features). Ces deux techniques sont intéressantes en raison de leurs bonnes performances et de leur faible coût. Les principales contributions de l'ORB sont les suivantes :

- L'ajout d'une composante d'orientation rapide et précise à FAST
- Le calcul efficace des caractéristiques BRIEF orientées
- Analyse de la variance et de la corrélation des caractéristiques BRIEF orientées
- Une méthode d'apprentissage pour la décorrélation des caractéristiques BRIEF sous invariance rotationnelle, conduisant à de meilleures performances dans les applications de voisinage.

1.2.1.1 FAST

Étant donné un pixel p dans un tableau, il compare rapidement la luminosité de p aux 16 pixels environnants qui sont dans un petit cercle autour de p .

Les pixels dans le cercle sont ensuite triés en trois classes (plus clair que p , plus sombre que p ou similaire à p). Si plus de 8 pixels sont plus sombres ou plus clairs que p , il est sélectionné comme point clé. Ainsi, les points clés trouvés par Fast nous donnent des informations sur

l'emplacement des bords déterminants dans une image.

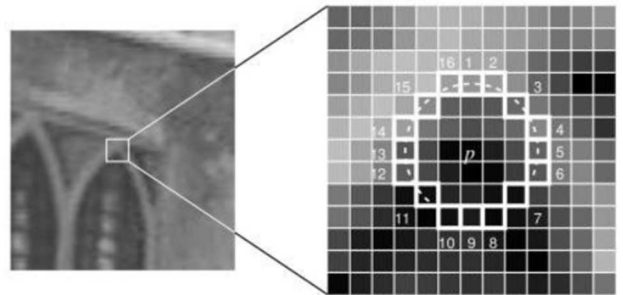


Figure 01 : *Voisinage de pixel*

Cependant, les caractéristiques de FAST n'ont pas de composante d'orientation et de caractéristiques multi-échelles.

L'algorithme ORB utilise donc une pyramide d'images à plusieurs échelles. Une pyramide d'images est une représentation multi-échelle d'une seule image, qui consiste en des séquences d'images qui sont toutes des versions de l'image à différentes résolutions. Chaque niveau de la pyramide contient la version de l'image qui est moins échantillonnée que le niveau précédent.

Une fois qu'ORB a créé une pyramide, il utilise l'algorithme FAST pour détecter les points clés de l'image. En détectant les points clés à chaque niveau, ORB localise effectivement les points clés à une échelle différente. De cette façon, ORB est partiellement invariant d'échelle.

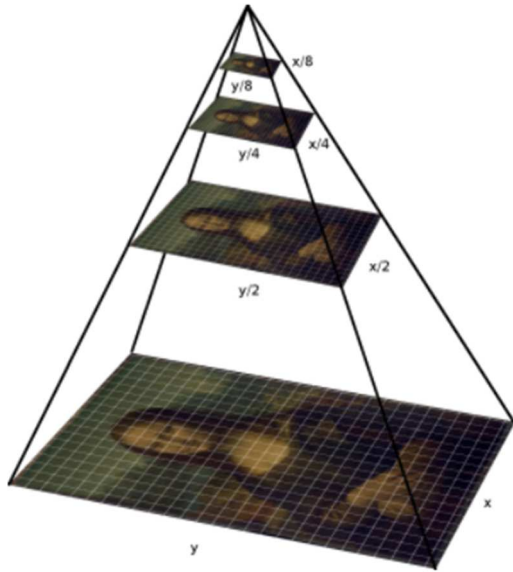


Figure 02 : *Pyramide de pixel*

Après avoir localisé les points clés, ORB attribue maintenant une orientation à chaque point clé, comme la gauche ou la droite, en fonction de la façon dont les niveaux d'intensité changent autour de ce point clé. Pour détecter les changements d'intensité, ORB utilise le centroïde d'intensité. Le centroïde d'intensité suppose que l'intensité d'un coin est décalée par rapport à son centre, et ce vecteur peut être utilisé pour imputer une orientation.

Tout d'abord, les moments d'un patch sont définis comme :

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (06)$$

Avec ces moments, nous pouvons trouver le centroïde, le "centre de masse" de la tache comme :

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (07)$$

Nous pouvons construire un vecteur du centre O de l'angle au centroïde -OC. L'orientation de la tache est alors donnée par :

$$\theta = \text{atan} 2(m_{01}, m_{10}) \quad (08)$$

1.2.1.2 BRIEF

BRIEF prend tous les points clés trouvés par l'algorithme rapide et les convertit en un vecteur de caractéristiques binaires afin qu'ils puissent représenter ensemble un objet.

Le vecteur de caractéristiques binaires, également appelé descripteur de caractéristiques binaires, est un vecteur de caractéristiques qui ne contient que des 1 et 0. En bref, chaque point clé est décrit par un vecteur de caractéristiques qui est une chaîne de 128 à 512 bits.

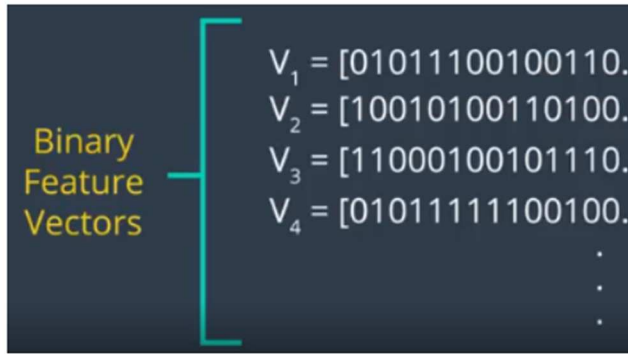


Figure 03 : Valeur Binaire des pixels

BRIEF n'est pas invariant à la rotation, donc ORB utilise rBRIEF (Rotation-aware BRIEF).

Considérons un patch d'image lissée, p . un test binaire τ est défini par :

$$\tau(p; x, y) = \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases} \quad (09)$$

où $p(x)$ est l'intensité de p en un point x . La caractéristique est définie comme un vecteur de n tests binaires :

$$f(n) = \sum_{1 \leq i < n} 2^{i-1} \tau(p; x_i, y_i) \quad (10)$$

La performance de correspondance de BRIEF chute fortement pour une rotation dans le plan de plus de quelques degrés. L'ORB propose une méthode permettant de diriger BRIEF en fonction de l'orientation des points clés. Pour tout ensemble de n tests binaires à l'emplacement (x_i, y_i) , nous avons besoin de la matrice $2 \times n$:

$$S = \begin{pmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{pmatrix} \quad (11)$$

Il utilise l'orientation des points clés θ et la matrice de rotation correspondante R_θ , et construit une version dirigée S_θ de S :

$$S_\theta = R_\theta S \quad (12)$$

Maintenant, l'opérateur BRIEF piloté devient :

$$g_n(p, \theta) = f_n(p) | (x_i, y_i) \in S_\theta \quad (13)$$

ORB est une fusion du détecteur de points clés FAST et de BRIEF avec quelques modifications.



Figure 04 : Exemples

1.3 SURF

H. Bay et al. ont présenté des caractéristiques robustes accélérées (SURF) en 2008, qui

repose également sur l'échelle gaussienne l'analyse spatiale des images. Le détecteur SURF est basé sur déterminant de la matrice de Hesse et il exploite les images intégrales pour améliorer la vitesse de détection des caractéristiques. Le descripteur 64 bin de SURF décrit chaque caractéristique détectée avec une distribution des réponses aux ondelettes de Haar dans certains quartiers. SURF sont invariables en termes de rotation et d'échelle, mais elles ont peu invariance affine. Toutefois, le descripteur peut être étendu à 128 valeurs de bin afin de traiter des changements de point de vue plus importants. L'équation représente le Matrice au point $x = (x,y)$ à l'échelle σ

Où $L_{xx}(x,\sigma)$ est la convolution de la seconde gaussienne

L'ordre dérivé avec l'image I au point x, et de même

Pour $L_{xx}(x,\sigma)$ et $L_{yy}(x,\sigma)$

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (14)$$

Le SURF repose sur le déterminant de la matrice de Hesse pour l'échelle et la localisation.

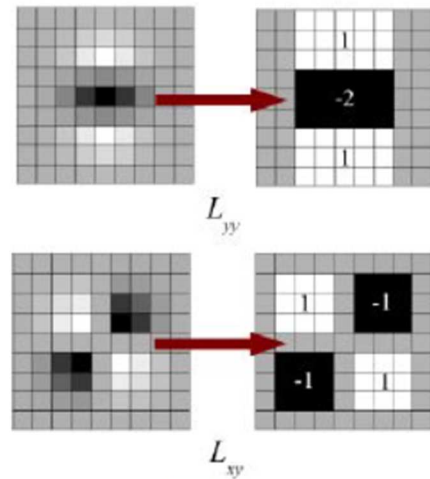


Figure 05 : Matrice de pixel

Pour l'affectation de l'orientation, le SURF utilise les réponses des ondelettes dans les directions horizontale et verticale pour un voisinage de taille 6. Des poids gaussiens adéquats lui sont également appliqués. Ensuite, ils sont tracés dans un espace comme indiqué dans l'image ci-dessous.

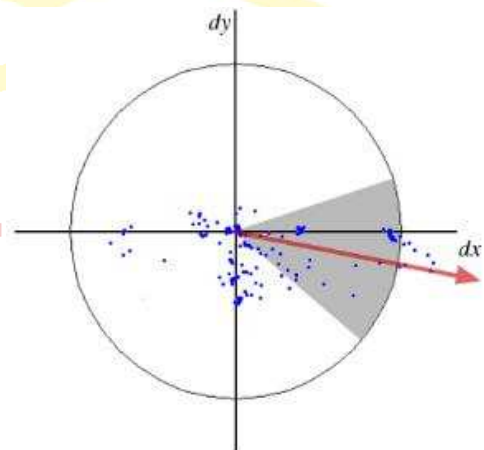


Figure 06 : Orientation de descripteur

Pour la description des caractéristiques, SURF utilise des réponses en ondelettes dans les directions horizontale et verticale (là encore, l'utilisation d'images intégrales facilite les choses). Un voisinage de taille 20 X20 est pris autour du point clé où s est la taille. Il est divisé en sous-régions 4x4. Pour chaque sous-région, des réponses en ondelettes horizontales et verticales sont prises et un vecteur est formé comme ceci

$$v = (\sum d_x, \sum d_u, \sum |d_x|, \sum |d_u|) \quad (15)$$

Lorsqu'il est représenté sous forme de vecteur, il donne le descripteur de caractéristique SURF avec un total de 64 dimensions.

Pour plus de spécificité, le descripteur de caractéristiques SURF a une version étendue à 128 dimensions.

Une autre amélioration importante est l'utilisation du signe de Laplacien (trace de la matrice de Hesse) pour le point d'intérêt sous-jacent. Le signe du Laplacien permet de distinguer les taches brillantes sur un fond sombre de la situation inverse. Lors de l'étape de comparaison, nous ne comparons les caractéristiques que si elles présentent le même type de contraste (comme le montre l'image ci-dessous). Cette information minimale permet une comparaison plus

rapide, sans réduire la performance du descripteur.

En résumé, SURF ajoute de nombreuses caractéristiques pour améliorer la vitesse à chaque étape.

2. Analyse des performances

Afin de mesurer les performances, de chaque algorithme de détection des points caractéristiques, nous allons suivre les étapes suivantes pour chaque détecteur:

- Analyser une image, détecter 50 points caractéristiques
- Attaquer l'image par différents attaques (bruits, étirement, compression, rotation, flou, découpage, filtre median, filtre moyenneur)
- faire l'analyse de l'image attaquée
- Faire correspondre les paires de caractéristiques entre l'originale et l'image attaquée.

Les scores de correspondances sont les indicateurs de performances, ainsi que le temps de détection de chaque algorithme.

Les images utilisées sont nommés de gauche à droite suivant: « bear », « papillon », « montre »



Figure 07 : Images test

2.1 Résultats Les résultats de correspondances ont été résumés dans ces graphiques :

Voici un aperçu d'une des images attaquées :

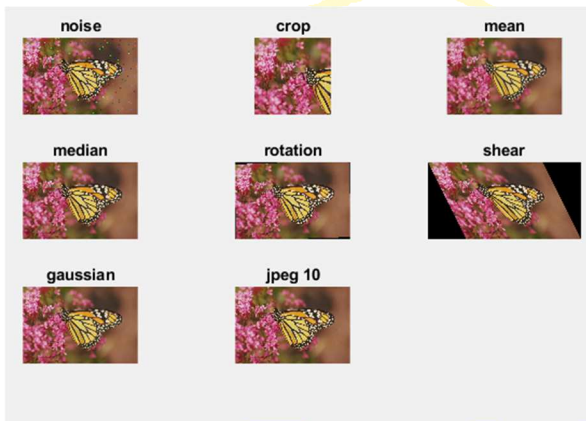


Figure 08 : Aperçu des attaques « papillon »

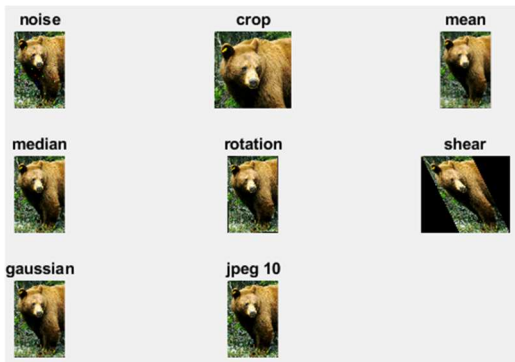


Figure 09 : Aperçu des attaques « bear »

Voici les graphes correspondant pour chaque image test.

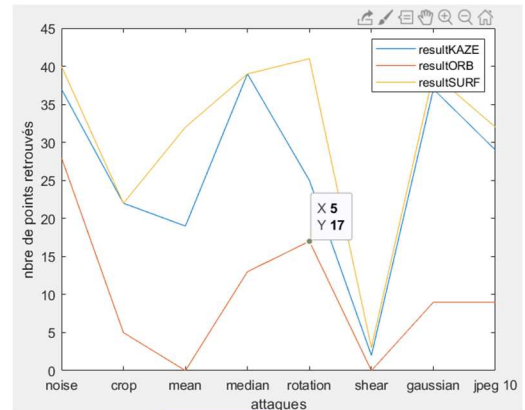


Figure 010 : Résultats bear

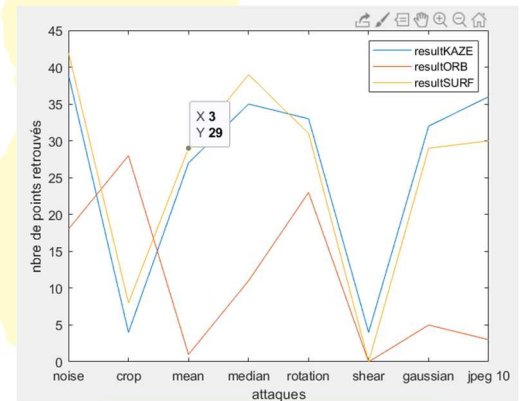


Figure 011 : Résultats papillons

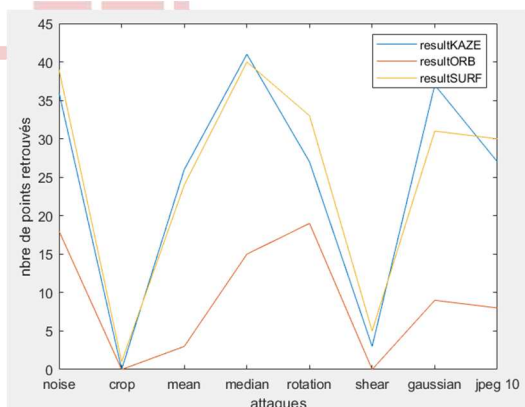


Figure 012 : Résultats montre

2.2 Interprétations:

Nous pouvons voir dans ces graphes les performances de chaque algorithme:

- Les caractéristiques KAZE sont invariantes à la rotation, à l'échelle, à l'affinité limitée et ont plus de spécificité à différentes échelles, au prix d'une augmentation modérée du temps de calcul.
- L'ORB est aussi performant pour la détection des caractéristiques (et est meilleur que le SURF) tout en étant presque deux ordres de grandeur plus rapide.
- SURF est bon pour traiter les images avec flou et rotation, mais pas pour traiter les changements de point de vue et d'illumination.

Analyse du temps d'exécution du descripteur :

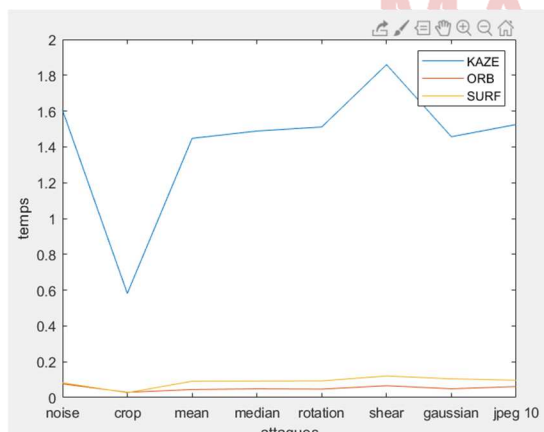


Figure 013 : Temps d'exécution montre

L'algorithme KAZE est moins performant en terme de vitesse de détection par rapport aux deux algorithmes ORB et SURF

3. Conclusion

Nous pouvons conclure qu'en termes de détection: SURF et KAZE se retrouvent dans une catégorie élevée par rapport à ORB.

Si nous parlons de stabilité, KAZE se trouve plus stable pour retrouver les descripteurs des images attaquées.

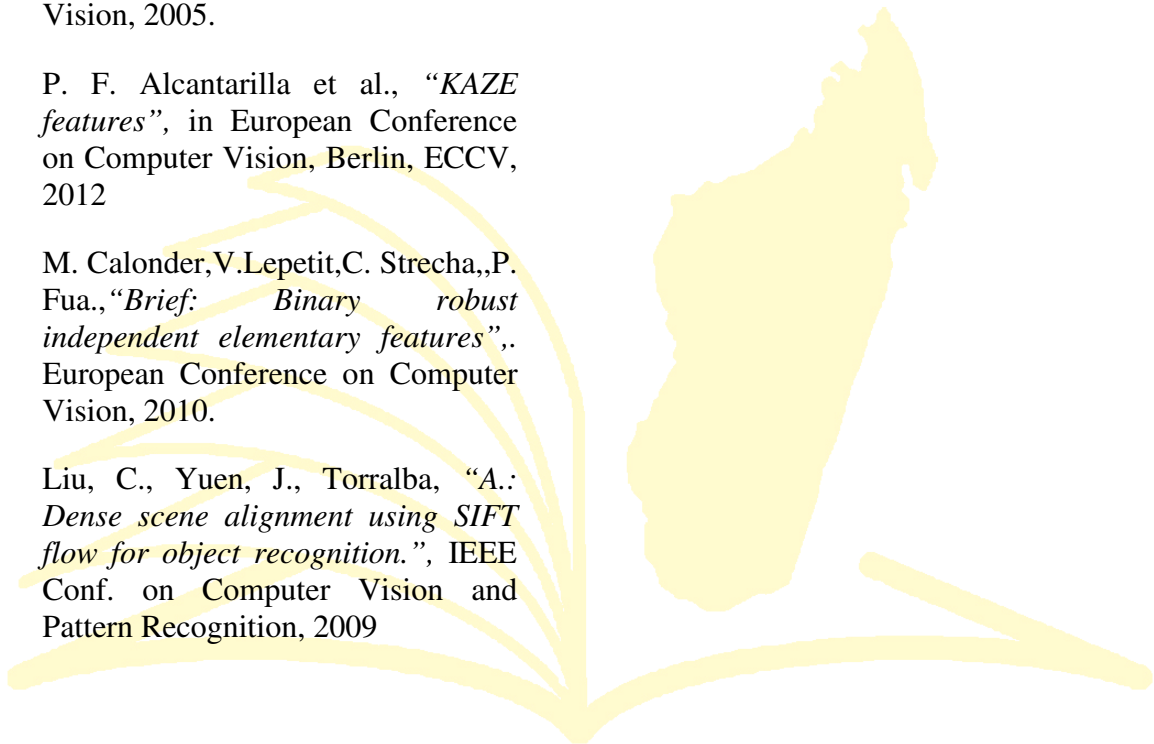
En analysant le temps d'exécution KAZE semble défavorisé par rapport aux deux autres.

Ainsi en fonction des objectifs et applications à atteindre pour les descripteurs, chaque algorithme a leur propre avantage et inconvénient.

BIBLIOGRAPHIE

- [1] M. Hassaballah et al., "Image features detection, description and matching." in Image Feature Detectors and Descriptors, Springer International Publishing, 2016.
- [2] S. Gauglitz et al., "Evaluation of interest point detectors and feature descriptors for visual tracking," International Journal of Computer Vision, 2011

- [3] D. G. Lowe, “*Distinctive image features from scale-invariant keypoints*”, International Journal of Computer Vision, 2004.
- [4] H. Bay et al., “*Speeded-up robust features (SURF)*,” Computer Vision and Image Understanding, 2008.
- [5] K. Mikolajczyk et al., “*A comparison of affine region detectors*”, International Journal of Computer Vision, 2005.
- [6] P. F. Alcantarilla et al., “*KAZE features*”, in European Conference on Computer Vision, Berlin, ECCV, 2012
- [7] M. Calonder, V. Lepetit, C. Strecha, P. Fua., “*Brief: Binary robust independent elementary features*”, European Conference on Computer Vision, 2010.
- [8] Liu, C., Yuen, J., Torralba, “*A.: Dense scene alignment using SIFT flow for object recognition.*”, IEEE Conf. on Computer Vision and Pattern Recognition, 2009



MADA-ETI