

# Detection de *features* et descripteurs

## Détecteur de Harris et algorithmes de non-suppression des maximas

Mohammed Amine KHELDOUNI

13 novembre 2018

L'objectif de ce travail est d'implémenter des algorithmes capable de détecter les *features* d'une image. Pour cela, on se propose dans un premier temps de trouver les coins grâce au détecteur de Harris permettant à partir des convolutions gaussiennes de l'image d'accéder aux nombreux descripteurs de coin perçus dans celle-ci (*corners*).

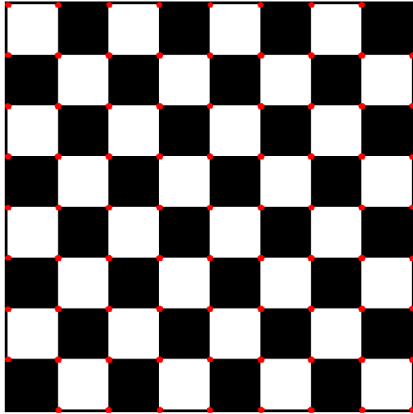
Toutefois, les *features* détectées par la méthode de Harris nécessite un affinement, car celle-ci renvoie un très grand nombre de descripteurs. Nous avons donc convenu d'utiliser un seuil (*threshold*) capable de filtrer certains scores trop faibles des descripteurs inintéressants.

Dans un second temps, nous avons implémenté deux algorithmes celui de suppression des non-maximas qui permet, étant donné un rayon  $r$ , de filtrer ces descripteurs selon les maximas locaux (dans la boule de rayon  $r$ ) pour améliorer les résultats.

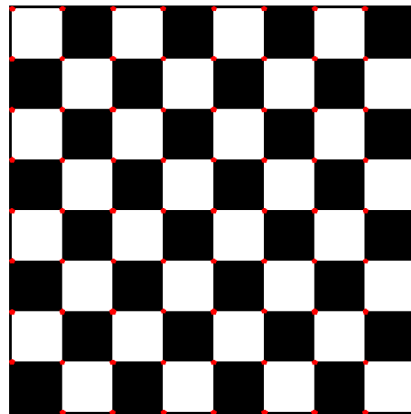
Enfin, nous avons élaboré un algorithme adaptatif de la suppression des non-maximas qui prend en compte la distribution des descripteurs sélectionnés et l'uniformise sur l'image pour avoir des descripteurs mieux répartis au sein de l'image.

## 1 Analyse des propriétés du détecteur

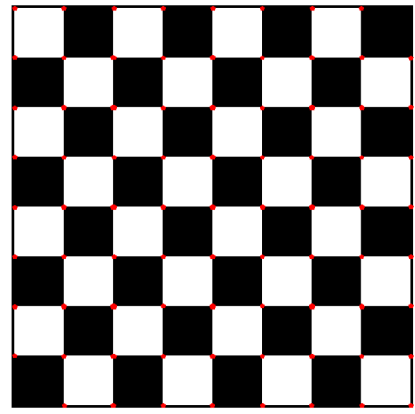
Dans un premier temps, nous avons validé nos trois algorithmes (*Harris Detector*, *Non-Maximum Suppression* et *Adaptive Non-Maximal Suppression*) sur une image représentant un échiquier. On remarque sur la figure ci-dessous que les coins sont bien recouverts par le détecteur de Harris pour lequel il a fallu calibrer le threshold  $\tau$ .



Points détectés par Harris sur  
un échéquier

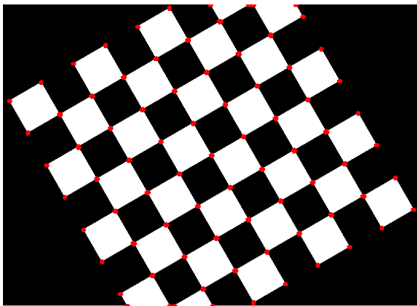


Points filtrés par l'algorithme  
NMS sur un échéquier

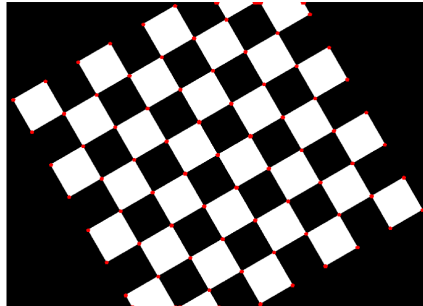


Points filtrés par l'algorithme  
ANMS sur un échéquier

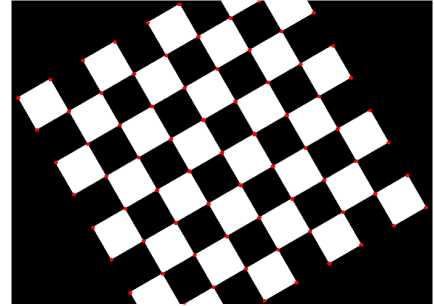
## 1.1 Sensibilité à la rotation



Points détectés par Harris sur  
un échéquier en rotation



Points filtrés par l'algorithme  
NMS sur un échéquier en  
rotation

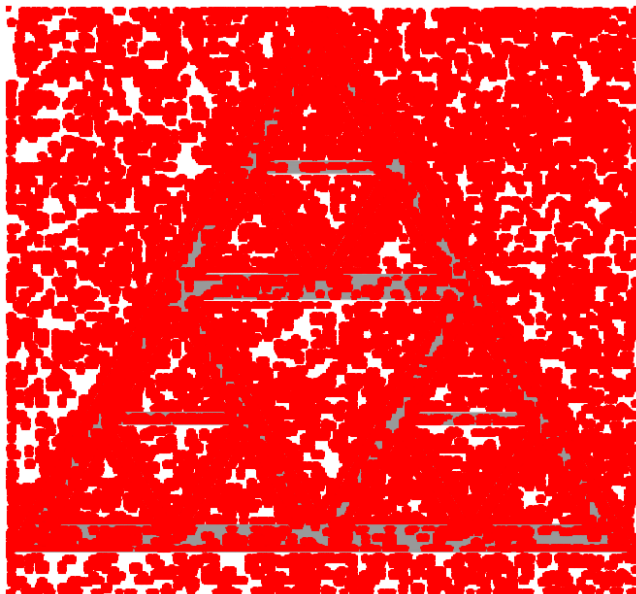


Points filtrés par l'algorithme  
ANMS sur un échéquier en  
rotation

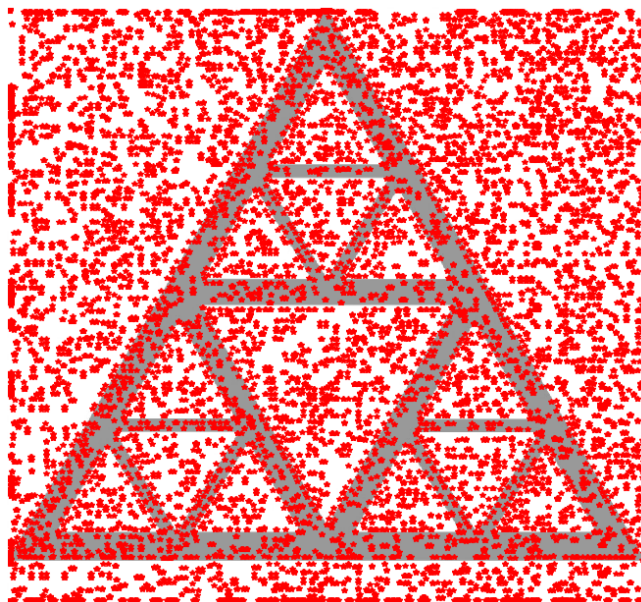
Nous retrouvons ici le fait que le détecteur est invariant par rotation mais aussi par homothétie (déformation de l'échelle ...) et par rapport au bruit tant que celui-ci ne fausse pas complètement les variations d'intensité de l'image. Nous avons également testé ces algorithmes sur le logo de l'Ecole Nationale des Ponts et Chaussées, sur une photo de peluche et sur la Tour Eiffel. Les résultats sont esquissés ci-dessous.

## 1.2 Threshold $\tau$

Nous remarquons qu'une faible valeur du threshold permet une plus grande tolérance vis à vis des points détectés. Comme l'algorithme ANMS est assez couteux en terme de calcul, nous avons calibré notre threshold pour qu'il puisse produire des points par le détecteur Harris assez nombreux, mais suffisamment filtrés par le *threshold* pour que l'algorithme NMS et ANMS retourne une solution dans un temps raisonnable. Ci-dessous un exemple où le détecteur de Harris a un threshold trop faible permettant de détecter plus de 300000 points.



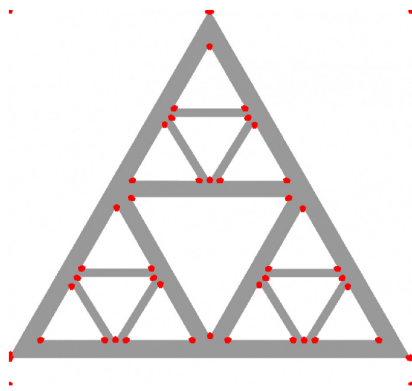
Points détectés par Harris (avec un faible threshold  $\tau$ ) sur le logo de l'ENPC



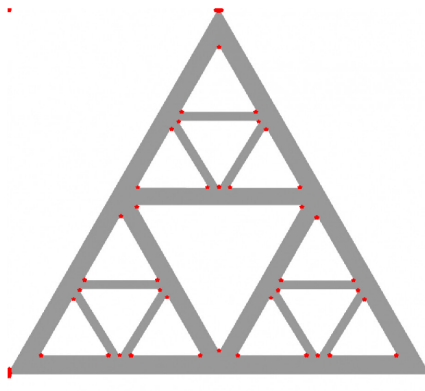
Points filtrés par l'algorithme NMS sur le logo de l'ENPC

## 2 Illustrations

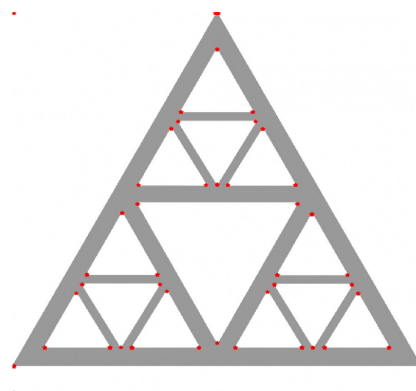
### 2.1 Exemple de détection sur le logo de l'Ecole des Ponts ParisTech



Points détectés par Harris sur le logo de l'ENPC

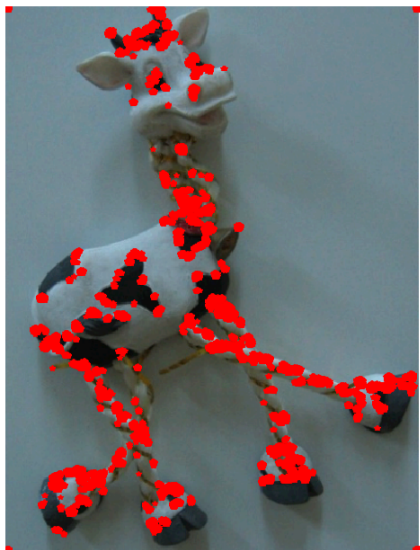


Points filtrés par l'algorithme NMS sur le logo de l'ENPC



Points filtrés par l'algorithme ANMS sur le logo de l'ENPC

## 2.2 Exemple de détection sur une figurine



Points détectés par Harris sur une figurine de vache



Points filtrés par l'algorithme NMS sur une figurine de vache



Points filtrés par l'algorithme ANMS sur une figurine de vache

## 2.3 Exemple de détection sur deux vue de la Tour Eiffel

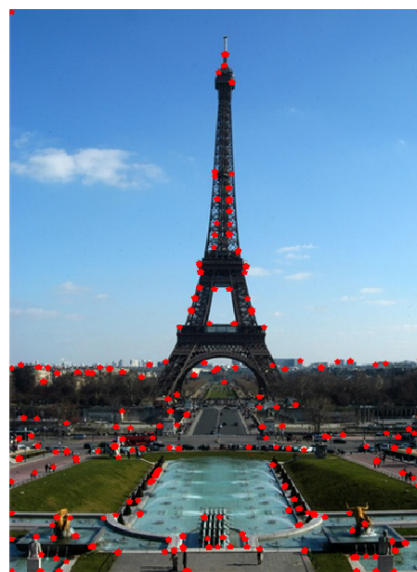
Un dernier résultat porte sur l'analyse d'une image de la Tour Eiffel prise de jour et de nuit (la perspective étant la même), pour comparer les détections des descripteurs sur l'une et l'autre des photos.



Points détectés par Harris sur la Tour Eiffel prise de jour



Points filtrés par l'algorithme NMS sur la Tour Eiffel prise de jour

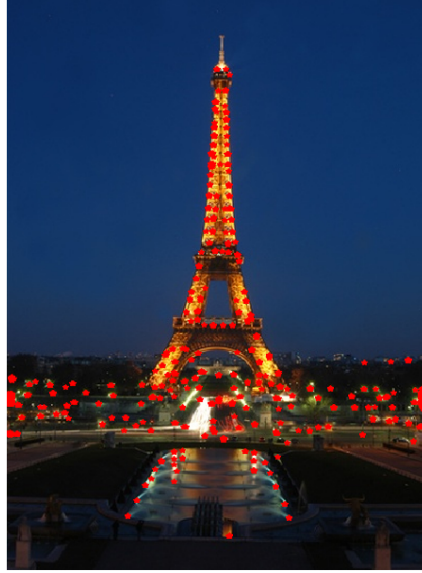


Points filtrés par l'algorithme ANMS sur la Tour Eiffel prise de jour

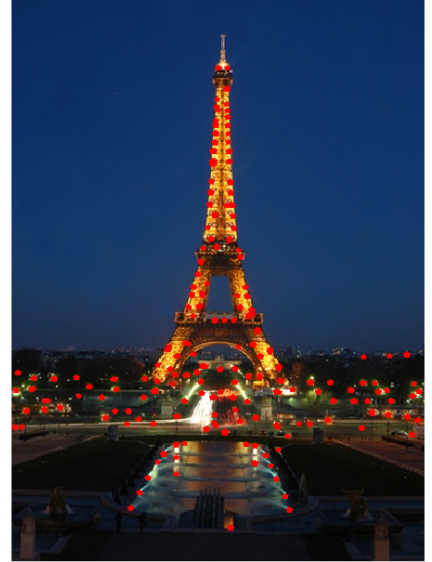




Points détectés par Harris sur la Tour Eiffel prise de nuit



Points filtrés par l'algorithme NMS sur la Tour Eiffel prise de nuit



Points filtrés par l'algorithme ANMS sur la Tour Eiffel prise de nuit

### 3 Commentaires

Nous remarquons que les trois algorithmes arrivent à détecter comme attendu des points remarquables dans les images traitées. L'algorithme basique du détecteur de Harris renvoie plusieurs détections, qu'il faut filtrer grâce à un threshold  $\tau$  et par une suppression des non-maxima locaux. Pour cela, nous utilisons les deux méthodes NMS et ANMS. On remarque que sur l'algorithme adaptatif, nous n'avons pas besoin de calibrer un rayon statique car celui-ci étant calculé à chaque points traité. L'ANMS palie donc au problème de distributivité des points détectés. En effet, la détection au plus uniforme et mieux filtrée grâce à l'algorithme adaptatif qui prend en compte les distances entre les différentes détections.

Enfin, sur les deux photos de la Tour Eiffel sur une même perspective à deux instants différents de la journée, nous remarquons que les points détectés sont globalement différents.

### 4 Conclusion

En somme, notre algorithme de détection des coins (*Harris Corner Detection*) permet d'esquisser une large panoplie de descripteurs potentiellement intéressants pour notre analyse de vision. Ces détections sont filtrées grâce à un seuil (*threshold*) et une suppression des non-maximums locaux. Nous avons amélioré cette suppression pour relacher l'hyperparamètre  $r$  (rayon qui définit le voisinage local d'un pixel) par une méthode adaptative qui filtre les points en leur affectant un rayon dynamique permettant une distribution plus uniforme des détections sur l'image.