

## Analyse d'images de pales d'éoliennes. Semaine d'étude maths-entreprises, Paris, 11-15 septembre 2017.

Anouk Nicolopoulos, Nizare Riane, Alexandre Saint-Dizier, Lara Trussardi

► **To cite this version:**

Anouk Nicolopoulos, Nizare Riane, Alexandre Saint-Dizier, Lara Trussardi. Analyse d'images de pales d'éoliennes. Semaine d'étude maths-entreprises, Paris, 11-15 septembre 2017.. [Rapport de recherche] AMIES. 2017, pp.1-18. hal-01651796

**HAL Id: hal-01651796**

**<https://hal.archives-ouvertes.fr/hal-01651796>**

Submitted on 29 Nov 2017

**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.

---

# ANALYSE D'IMAGES DE PALES D'ÉOLIENNES

*par*

Alexandre Saint-Dizier, Anouk Nicolopoulos, Nizar Riane & Lara Trussardi

---

**Résumé.** — L'entreprise Cornis est spécialisée dans la photographie haute-résolution de pales d'éoliennes pour le travail de maintenance. Dans le cadre de la Semaine d'Étude Maths-Entreprises (SEME) organisée par le MAP5 de l'Université Paris Descartes du 11 au 15 septembre 2017, nous avons tenté de donner des réponses à deux problématiques soulevées par l'entreprise et concernant l'analyse de ces images.

## Table des matières

1. Introduction.....	2
2. Analyse des niveaux de gris de l'image.....	5
3. Détection de la pale.....	9
4. Réponse au premier problème.....	13
5. Réponse au second problème.....	15
6. Problèmes et idées à développer.....	16

## 1. Introduction

La production d'énergie éolienne atteint aujourd'hui 960TWh dans le monde, et prend de plus en plus d'importance. Des champs d'éoliennes sur terre ou en mer sont installés, avec des pales longues de 40 à 80m. La maintenance des ces pales est souvent effectuée par un technicien envoyé sur place. Celui-ci monte en haut du mât de l'éolienne pour y observer à l'oeil nu les défauts, encordé ou soutenu par une plateforme qui fait le tour de la pale. Mais cela prend de l'ordre d'un jour et demi par éolienne.

Cornis a donc développé un outil, appelé *Panoblade*, qui permet de réduire le temps passé sur une éolienne à une heure et demie en numérisant les pales de l'éolienne. Cela permet de n'envoyer un technicien réparer la pale en haut du mât qu'en cas où un défaut a été repéré.

**1.1. Acquisition des photos.** — L'objectif est de pouvoir déterminer avec précision l'état des pales en analysant les scans sur l'écran d'un ordinateur.



FIGURE 1. Installation.

L'appareil photo est placé au pied de l'éolienne et prend une cinquantaine de photos par face de pale. Pour cela, l'appareil est fixé sur une tête motorisée qui lui permet de suivre du bout d'une pale au hub de l'éolienne puis de celui-ci au bout de la seconde pale. On incline ensuite les pales, et on recommence pour avoir les deuxièmes faces. Puis on fait tourner l'éolienne de 120 degrés et on recommence.

Ce travail prend entre 1h et 1h30, et on comprend que cette technique est plus avantageuse que de faire monter une personne en haut de l'éolienne.

**1.2. Premier problème : détection de photos floues.** — Il arrive que des photos soient trop floues pour pouvoir repérer des défauts. Cela arrive notamment lorsqu'il y a un problème de mise au point, quand l'appareil repère des nuages par exemple. Ou encore lorsqu'il s'agit d'une photo du hub de l'éolienne. Voir la figure 2.

Dans ces cas-là, on voudrait que le logiciel détecte les photos floues et qu'il relance l'appareil pour reprendre les photos directement sur le terrain.



FIGURE 2. Exemples de photos floues. Pale sur fond de nuages à gauche, hub à droite.

**1.3. Second problème : qualité des photos.** — Ce problème est moins ciblé que le premier. On cherche une mesure de qualité des photos. En étudiant la base de données des photos de pales de Cornis, cette mesure permettrait de comprendre dans quelles conditions les photos sont le plus satisfaisantes.

**1.4. Point de départ de la réflexion.** — Une méthode classique de détection de flou sur une image consiste à analyser son spectre. On décompose l'image en signaux, et on regarde son périodogramme, qui est défini pour une fréquence  $\omega$  par

$$P(\omega) = \frac{1}{N} \left| \sum_{n=1}^N x(n) \exp^{-i\omega n} \right|^2$$

où les  $x(n)$  sont les signaux de l'image. Le périodogramme est censé être aplati dans le cas d'une photo floue par rapport à une photo nette, voir figure 3. Mais la différence n'est pas si flagrante, bien que les oscillations soient moins amples pour la photo floue, à gauche.

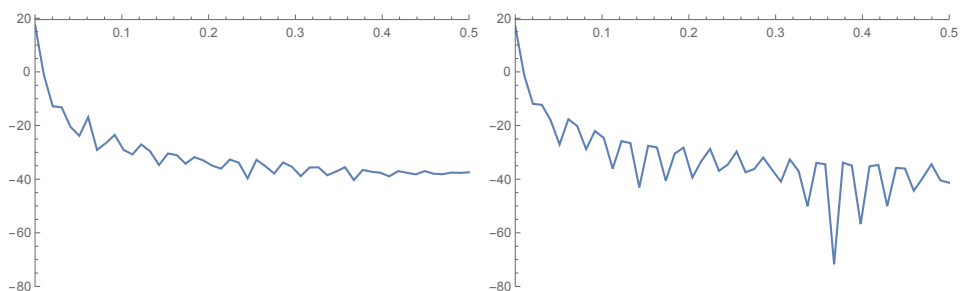


FIGURE 3. Périodogrammes d'une photo floue, à gauche, et d'une photo nette, à droite.

Étant donné le bruit, la diversité de points de vue et de type d'éclairage des images, il est difficile de segmenter les images de manière robuste et efficace.

Les photos de pales sont a priori assez simples, composées d'un bout de pale rectiligne et de ciel. Comme il arrive qu'une photo soit nette mais que la pale soit lisse et le ciel uniforme, cela empêche d'analyser algorithmiquement et même à l'oeil nu la netteté de l'image à partir de la pale seulement.

Nous avons donc essayé d'évaluer le niveau de flou à partir de la transition ciel-pale.

On a donc commencé par regarder ce que l'on obtient lorsqu'on applique un filtre laplacien. Il s'agit d'un filtre classique de détection de contour. Il repère tous les pixels  $(x, y)$  où le niveau de gris  $v(x, y)$  est tel que

$$v(x + 1, y) + v(x, y + 1) - 4v(x, y) + v(x, y - 1) + v(x - 1, y) = 0$$

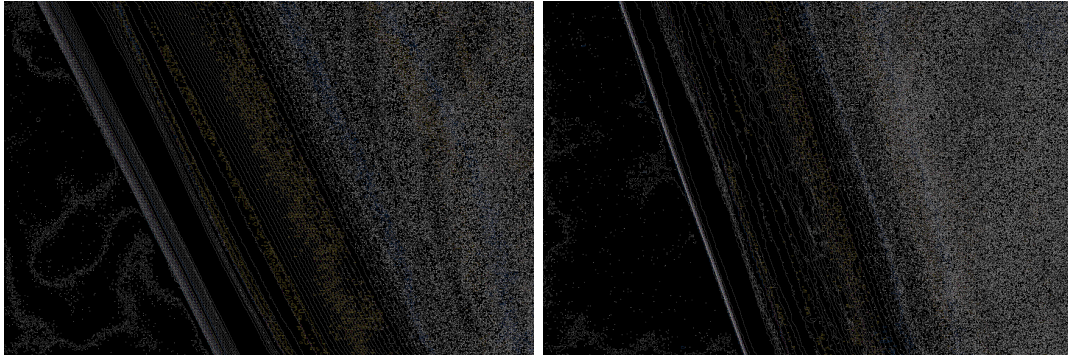


FIGURE 4. Filtre laplacien appliqué à une photo floue, à gauche, et à une photo nette, à droite.

Mais on peut voir sur la figure 4 que l'application de ce filtre à une image floue (à gauche) et à une image nette (à droite) ne donne pas de résultats très différents, ni d'information très précise.

Nous avons donc analysé directement les niveaux de gris des images fournies par Cornis.

Cependant la pale peut être plus ou moins large, inclinée de n'importe quel angle par rapport à l'horizontale, le fond peut être uniforme ou comporter des nuages... Et quand il s'agit du bout de la pale, il n'est souvent présent que sur la moitié de la photo.

Après avoir supposé que l'on travaillait sur une ligne de pixels qui traversait la frontière ciel-pale pour déterminer un critère de tri entre photos nettes et photos floues dans la section 2, nous avons donc cherché à détecter la pale dans la section 3.

Les résultats sont présentés dans la section 4 pour le premier problème, dans la section 5 pour le second problème.

Enfin, nous avons fait une liste de cas pour lesquels notre approche ne fonctionne pas et proposé quelques pistes pour résoudre ces problèmes.

## 2. Analyse des niveaux de gris de l'image

L'analyse des niveaux de gris de l'image permet de bien visualiser les transitions ciel-pale. On rappelle qu'on a supposé que l'on travaillait sur une ligne qui traverse l'interface ciel-pale. Comme on peut le voir en figure 5, dans le cas d'une photo nette (courbe rouge), on observe un saut, et dans le cas d'une photo floue (courbe bleue), on observe une pente très marquée. On va donc s'intéresser à cette pente, pour voir si on peut trouver un seuil sous lequel rejeter les photos.

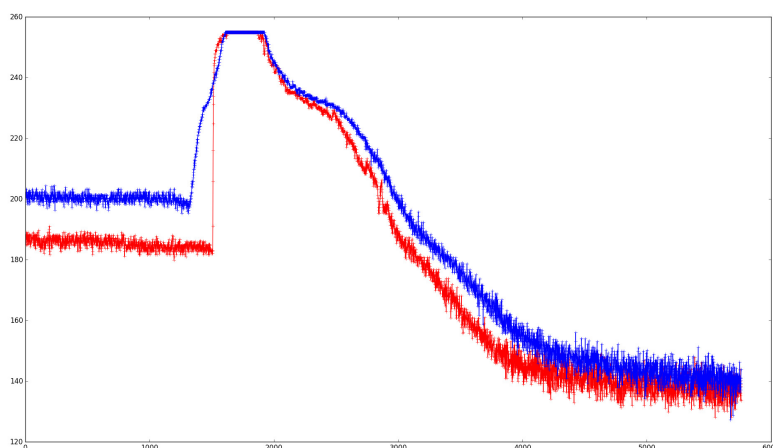


FIGURE 5. Exemple d'évolution des niveaux de gris sur une ligne traversant l'interface ciel-pale. Pour une photo floue en bleu, pour une photo nette en rouge. Ces deux photos ont été prises d'affilée et ont des caractéristiques très proches.

**2.1. Différence de niveaux de gris à pas  $k$  fixé.** — L'idée est d'enregistrer pour chaque photo le maximum de la différence de niveaux de gris entre les pixels éloignés d'un pas  $k$ . On regarde donc le vecteur contenant les niveaux de gris de chaque pixels

$$\begin{array}{|c|c|c|c|c|} \hline i & i+1 & \dots & i+k & i+k+1 \\ \hline \end{array}$$

et on calcule

$$\max_i |v(i) - v(i+k)|$$

où  $v(i)$  correspond au niveau de gris du pixel  $i$ .

Les photos fournies par Cornis ont deux caractéristiques particulières :

- parmi l'ensemble des photos d'une face de pale, il y a toujours des photos nettes,
- la mise au point est effectuée toutes les 5 photos donc quand il y a des photos floues cela concerne souvent un groupe de photos prises d'affilée.

De plus, elles sont triées en dossiers dont chacun contient une soixantaine de photos associées au scan d'une face de pale.

Lorsqu'on s'intéresse à un dossier de photos d'une face de pale, s'il y en a des floues elles auront donc nécessairement une pente bien moins raide que celle de la meilleure photo. Dans la suite, nous nous sommes fixés un écart de  $k = 2$ .

En regardant les meilleures pentes de chaque photo d'un dossier normalisées par rapport à la plus grande pente, on voit que souvent les photos floues sont sous le seuil 0.2, comme dans la figure 6.

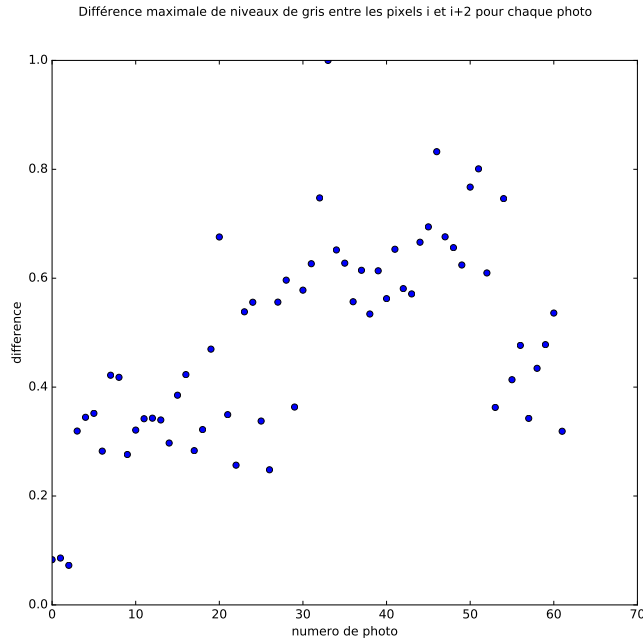


FIGURE 6. Différence maximale de niveaux de gris entre les pixels  $i$  et  $i + 2$  pour chaque photo d'un dossier.

Après avoir regardé les résultats pour quelques dossiers ayant des caractéristiques différentes (pales différentes, ciel bleu, blanc ou nuageux), le seuil 0.2, correspondant à

$$\max_i |v^n(i) - v^n(i + k)| \leq \frac{2}{10} \max_m \max_i |v^m(i) - v^m(i + k)|$$

où cette fois-ci  $v^n$  correspond au niveau de gris de l'image  $n$ , permettrait de séparer les photos floues des photos nettes.

Cette méthode permet donc de faire un tri assez grossier. Pour pousser plus loin la réflexion sur cette différence de niveaux de gris entre deux pixels éloignés d'un certain pas  $k$ , on a ensuite fait varier ce pas.

**2.2. Analyse en fonction du pas  $k$ .** — Dans la section précédente, nous avons regardé la différence maximale de niveaux de gris entre deux pixels ayant un certain écart. Ce qui va nous intéresser ici est justement cet écart plus que la valeur du maximum. Ainsi, au lieu de

considérer une valeur d'écart arbitraire, nous en considérons plusieurs, jusqu'à détecter l'écart qui nous intéresse : on l'appellera la longueur de la transition.

On s'intéresse donc cette fois-ci au

$$\max_k \max_i |v(i) - v(i + k)|.$$

Pour une image floue, autour de l'interface, la différence maximale entre deux pixels consécutifs sera faible et augmentera linéairement jusqu'à ce que l'écart dépasse la taille de la zone de transition. Ensuite cette différence sera constante, ne pouvant dépasser la différence entre la couleur de la pale et du ciel. Ainsi, pour détecter l'écart voulu, il suffit de regarder la courbe de la différence maximale en fonction de l'écart, et de sélectionner le point où la courbe devient constante.

Expérimentalement, cette hypothèse semble être vérifiée, comme l'illustre la figure 8. Mais comme on l'a précisé, elle n'est valable que localement. Il peut se passer plus de choses loin de l'interface, s'il y a une seconde transition de la pale au ciel, s'il y a des nuages, s'il y a beaucoup d'ombre ou encore si on est vers le hub, voir la photo 3 de la figure 8. Il y a donc beaucoup de détails à optimiser pour s'assurer un algorithme robuste, comme la zone considérée autour de l'interface, et la méthode de détection du point de transition à partir de la courbe des différences de niveaux de gris en fonction de l'écart.

On commence par calculer la courbe, puis on la *penche* en soustrayant la droite passant par le premier et le dernier point. On s'attend à avoir une sorte de fonction chapeau, avec le point de transition ciel-pale qui correspond au maximum.

Il faudrait maintenant s'assurer que la courbe ait bien la forme voulue, et que la largeur de la bande considérée recouvre bien la transition ciel-pale.

Nous avons estimé que le résultat était satisfaisant lorsque :

- le maximum de la courbe penchée était supérieur à la moitié du maximum de la courbe initiale, ce qui permet de s'affranchir des cas où la courbe est encore quasi-linéaire et où la bande ne recouvre pas totalement la transition,
- le maximum était supérieur à un certain seuil, ce qui permet de s'affranchir des cas où du bruit perturbe la courbe au début.

Nous commençons donc avec un intervalle de quelques pixels dont on augmente la taille jusqu'à ce que le maximum ait la forme convenue, voir l'algorithme 1.

Sur la figure 8, on peut voir les résultats obtenus pour les deux photos de la figure 7. La longueur de transition est de 250 pixels à gauche pour la photo 1, floue. Elle est de 15 pixels au centre pour la photo 2, nette. , Sur la figure 9 on voit le résultat pour une photo de hub d'éolienne, il s'agit du graphique complet que l'on peut obtenir sans faire attention à l'échelle. On remarque que le point d'intérêt est quand même détectable et correspond à un écart de 5 : l'image est nette.



**Algorithme 1** Algorithme d'estimation de flou**Entrée:** Image  $I$ **Sortie:** Mesure de flou

```

1: Calculer l'angle  $\alpha$  de la pale, et le point  $P$  de l'interface
2: Extraire la ligne  $l$  repérée par  $(\alpha, P)$ 
3: for  $n \in [0..n_{Max}]$  do
4:   DiffMax[n]  $\leftarrow \max_i l[i+n] - l[i]$ 
5: end for
6: range  $\leftarrow 10$ 
7: while True do
8:   ecart  $\leftarrow \arg \max_{i \in [0..range]} \text{DiffMax}[i] - \frac{i \times \text{Diffmax}[range]}{range}$ 
9:   if ecart non satisfaisant then
10:    range  $\leftarrow \text{range} + 1$ 
11:   else
12:    Return ecart
13:   end if
14: end while

```

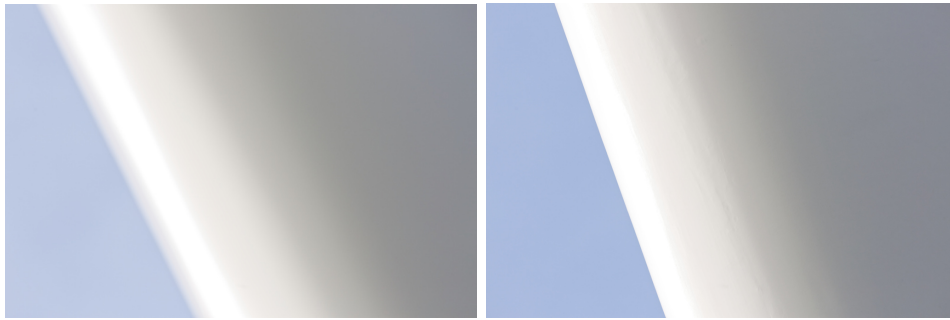


FIGURE 7. Exemples de photos. De gauche à droite, photos 1, 2.

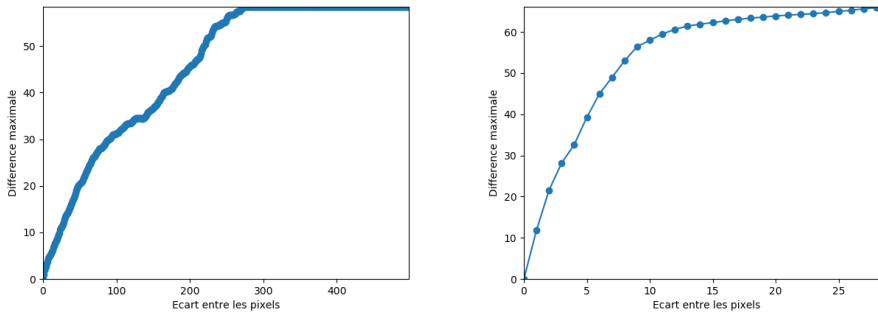


FIGURE 8. Graphiques correspondants obtenus.

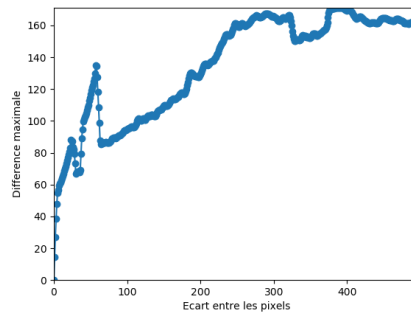


FIGURE 9. Graphique obtenu pour un hub d'éolienne.

### 3. Détection de la pale

Maintenant que l'on sait comment analyser la transition ciel-pale, il faut détecter cette ligne sur chaque photo. En pratique, nous allons chercher une droite perpendiculaire à la pale et un point de l'interface ciel-pale par lequel faire passer cette droite.

L'image étant très bruitée, nous avons choisi de réduire la qualité et de la flouter pour s'affranchir de ce bruit sans pour autant perdre les informations qui nous intéressent. En effet, cela nous fait perdre certains détails, mais permet aussi de calculer plus rapidement l'angle et ce plus efficacement puisqu'on est moins sensibles au bruit. Une fois la pale détectée, on retravaillera sur les photos initiales pour les différences de niveaux de gris.

**3.1. Détection de l'angle.** — La méthode consiste à étudier le gradient de l'image en forme polaire. Statistiquement, le gradient a une orientation uniforme, sauf à l'interface ciel-pale (voire même sur la pale, s'il y a une zone d'ombre par exemple), où il sera majoritairement orienté perpendiculairement à la pale. A priori, on devrait donc observer un pic sur l'histogramme des angles des gradients de l'image lorsque celui-ci correspond à l'angle de la perpendiculaire à la pale.

Pour faire encore plus abstraction du bruit, on ne garde que les points dont on considère que le gradient a une norme suffisante, à savoir supérieure à la valeur moyenne. En effet, on peut redouter la présence de nombreux points associés au bruit, qui auront tous une norme de gradient similaire. Prendre la moyenne comme seuil permet d'éliminer ces outliers en s'assurant de ne pas perdre les valeurs importantes. La méthode est mise en oeuvre dans l'algorithme 2.

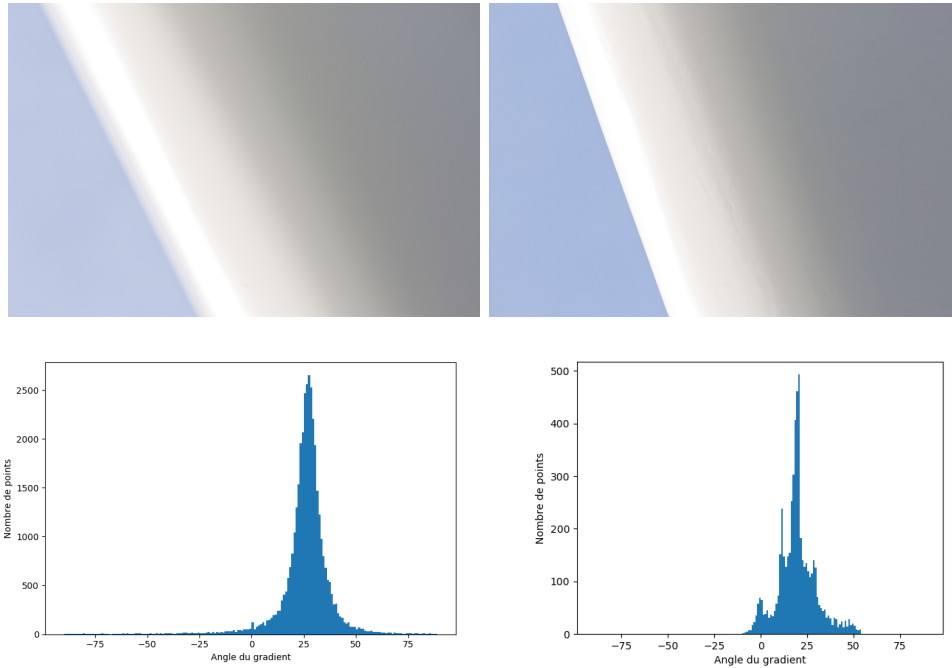


FIGURE 10. Histogramme des angles pour les photos 1 et 2.

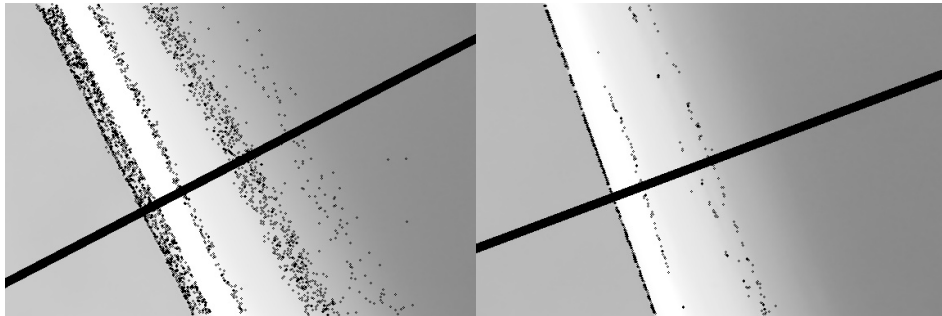


FIGURE 11. Points de gradient conservés et angles détectés (lignes en gras) pour les photos 1 et 2.

Comme on peut le voir sur la figure 10, l'histogramme obtenu correspond bien à ce qu'on attendait. Le pic n'est pas concentré en un angle, voir figure 11, puisque

- l'ombre sur la pale, qui est arrondie, fait apparaître des lignes pas tout à fait parallèles à celle que l'on recherche,
- la pale n'est en pratique pas parfaitement droite,
- il reste du bruit sur l'image.

Cependant, cela suffit pour détecter l'angle de la pale dans la grande majorité des cas de figures.

---

**Algorithme 2** Algorithme de calcul de l'angle de la pale.

---

**Entrée:** Image sous-résolue et floutée  $I$  ( $300 \times 500$ )**Sortie:** Angle  $\alpha$  perpendiculaire à l'axe de la pale

- 1: Calculer le gradient en forme polaire :  $\theta, |\Delta I|$
  - 2: On ne garde que les points dont la norme du gradient est supérieure à la moyenne
  - 3:  $\theta \leftarrow \theta[\Delta I > \bar{\Delta I}]$
  - 4: Histogramme  $\leftarrow$  histogramme( $\theta$ )
  - 5:  $\alpha = \arg \max$  Histogramme
- 

**3.2. Détection de la frontière ciel-pale.** — Toujours en se basant sur l'observation que la pale est essentiellement rectiligne, on peut espérer détecter les frontières, ou au moins l'une des deux potentielles dans les cas où on peut voir les deux bords de la pale sur la photo. En effet, les points du contour ont a priori un gradient ayant le même angle que celui calculé dans le paragraphe précédent. Nous pouvons donc restreindre nos recherches à une droite ayant la bonne orientation à partir des points ayant un gradient bien orienté et de norme suffisante.

Cela réduit le problème à la détection d'une droite parmi un ensemble de points comprenant beaucoup de points sur cette droite que l'on recherche et beaucoup d'outliers. Pour cela, une méthode est d'utiliser un *Random Sample Consensus* (RANSAC). L'idée est de prendre un couple de points au hasard, de faire correspondre une droite à ces points, et de regarder combien de points sont situés sur cette droite. On définit son score comme étant ce nombre de points. L'algorithme réitère le processus un certain nombre de fois, et sélectionne la ou les droites avec le meilleur score.

En pratique, puisque nous avons accès à l'angle que forme la droite avec l'horizontale, cela va permettre d'itérer sur chaque point, et de garantir une implémentation linéaire ou quasi-linéaire en éliminant petit à petit les points déjà observés, voir l'algorithme 3.

---

**Algorithme 3** Algorithme de détection de l'interface
 

---

**Entrée:** Ensemble  $E$  de points avec le bon gradient, angle de la pale  $\alpha$

**Sortie:** Droite  $\mathcal{D}^*$  la plus représentée

- 1: **for** point  $P$  dans  $E$  **do**
  - 2:      $\mathcal{D} \leftarrow$  Droite passant par  $P$  et d'angle  $\alpha$
  - 3:      $score(\mathcal{D}) = |\{E \cap \mathcal{D}\}|$
  - 4: **end for**
  - 5:  $\mathcal{D}^* = \arg \max_{\mathcal{D}} score(\mathcal{D})$
- 

Comme on peut le voir sur la figure 11, l'hypothèse selon laquelle la plupart des points se trouvent sur l'interface ciel-pale semble être justifiée.

Finalement, on détermine une droite traversant la frontière ciel-pale en récupérant les points qui sont sur l'interface et en faisant la moyenne, voir figure 12.

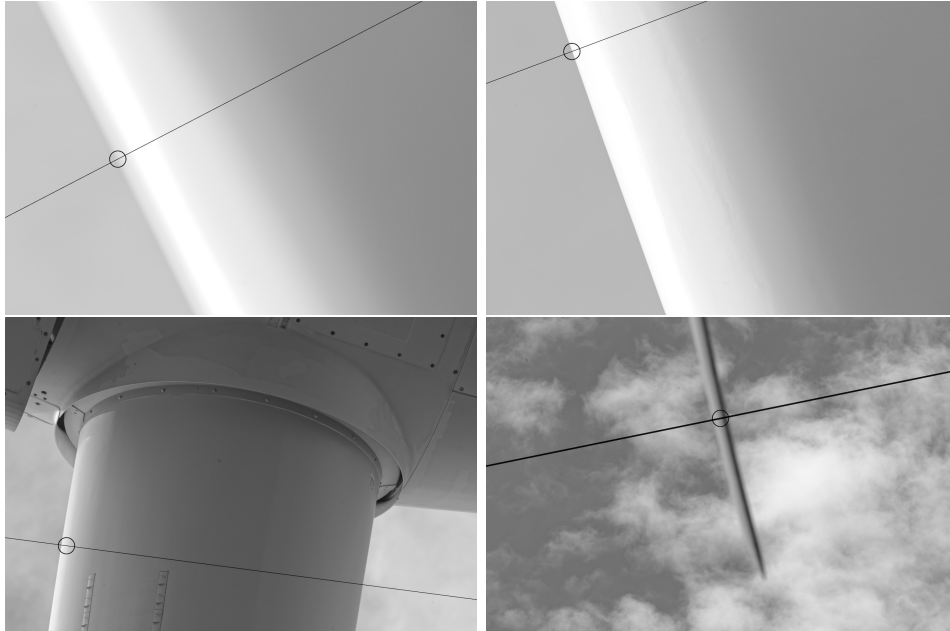


FIGURE 12. Résultat du calcul de la droite perpendiculaire à la pale et passant par l'interface ciel-pale pour plusieurs photos.

#### 4. Réponse au premier problème

On rappelle que l'objectif était de trouver une méthode pour détecter automatiquement les photos avec un gros flou de mise au point.

La méthode décrite dans le paragraphe 2.1, où l'on trie donc les photos ayant une pente maximale trop petite par rapport à la pente maximale de toutes les photos du dossier donne les résultats suivants

résultat exact	10
à 1 ou 2 photos près	9
à 8 photos près	6
très mauvais	2

FIGURE 13. Test sur 27 dossiers avec la méthode à pas fixe.

pour un pas  $k = 2$  entre les pixels considérés et un seuil de 0.2 comme critère de rejet des photos.

Les deux cas pour lesquels le résultat était très mauvais correspondent à des dossiers où les photos étaient peu floues. Celui-ci était d'ailleurs à détecter à l'oeil nu, et notre code nous a indiqué que la quasi-totalité des photos étaient *floues* (selon son critère).

Avec la seconde méthode, décrite dans le paragraphe 2.2, la séparation entre photos nettes et photos floues est plus marquée (figure 14) que dans le cas de la première méthode (figure 6). Nous avons fixé le seuil de rejet à 50 pixels.

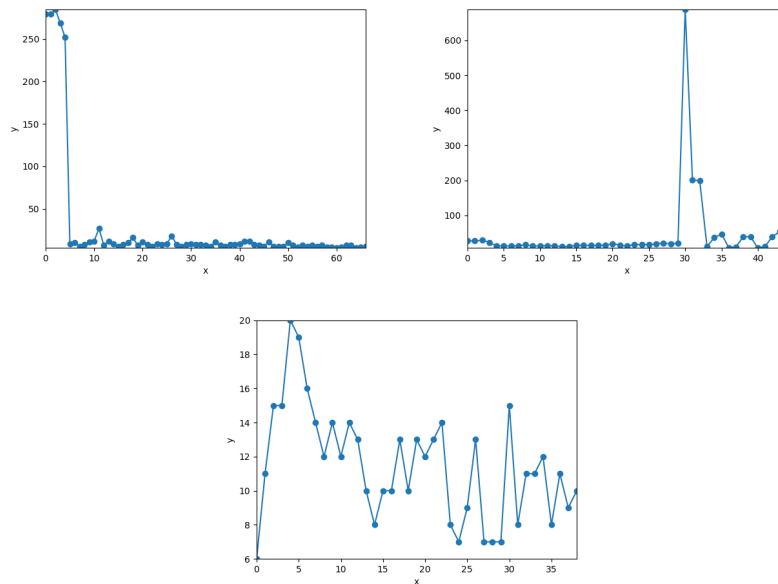


FIGURE 14. Exemples de graphique obtenu sur plusieurs dossiers. Ici les photos floues ont toutes une transition plus grande que 100 pixels.

Nous avons lancé le code sur 25 dossiers. Les erreurs que l'on a relevé semblent témoigner d'une faille dans les détails de l'implémentation plutôt que dans l'idée générale de l'algorithme :

- la plupart des faux positifs ont une longueur de transition supérieure à 1000 pixels, alors que les photos floues ont d'habitude une longueur de transition de l'ordre de la centaine de pixels,
- la plupart des faux négatifs, qui sont très rares, correspondent à des photos légèrement floues et dont la transition mesure quelques pixels de moins que notre seuil de 50 pixels.

résultat exact	11
à 1 ou 2 photos près	4
à 8 photos près	7
très mauvais	5

FIGURE 15. Test sur 25 dossiers avec la méthode de mesure de la transition.

## 5. Réponse au second problème

La première problématique était d'arriver à différencier les images floues des images nettes, et on voulait ensuite mettre au point un estimateur de flou de l'image.

La première méthode de tri à partir d'un seuil arbitraire est trop grossière pour vraiment mesurer la qualité des images.

La seconde approche, avec le calcul de la longueur de transition ciel-pale, est cette fois-ci indépendante du contraste de l'image et permet de définir un critère intuitif : la photo est floue si cette transition est longue. C'est donc une piste de réponse au second problème.



## 6. Problèmes et idées à développer

**Mauvaise détection de l'angle.** Pour la détection de l'angle de la normale à la pale, l'exemple de la figure 16 est caractéristique des photos de hub, et correspond à l'un des rares cas où l'angle est mal estimé, avec les photos sans ciel.

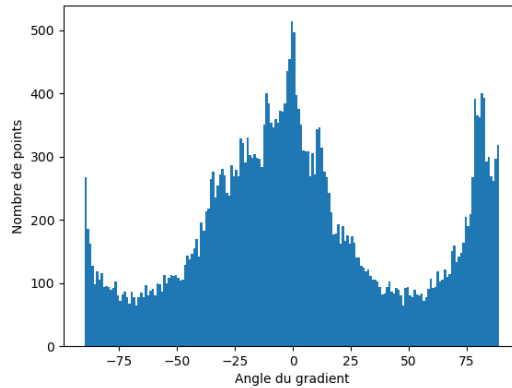


FIGURE 16. Exemple de photo de hub, cas où l'angle est mal calculé.

Ceci dit on peut voir sur l'histogramme qu'il y a deux pics, et moins marqués que dans les cas des photos 1 et 2, voir figure 10. On pourrait essayer de mesurer la confiance de cet estimateur d'angle pour avoir un algorithme plus robuste.

**Mauvaise détection de la frontière ciel-pale : problème de détection de la droite.** Pour la détection d'une normale à la pale traversant la frontière ciel-pale, les résultats obtenus correspondent à nos attentes dans les cas où l'angle a été correctement estimé, sauf dans le cas où il n'y a presque pas de ciel, voir figure 17. L'algorithme pourrait être amélioré en détectant les deux meilleures droites puis en les confrontant, par exemple, pour garder la droite qui a la pente de niveaux de gris la plus élevée.

**Mauvaise détection de la frontière ciel-pale : problème de détection du point sur la frontière.** Il y a aussi les cas où la droite est correcte mais pas la localisation de la frontière ciel-pale. La figure 18 est un exemple où l'algorithme se trompe à cause de la confusion de la transition ciel-pale avec la transition ombre-lumière. Comme on utilise aussi le point détecté par l'algorithme 3 pour faire une étude locale, la mauvaise détection de ce point pose problème. Ce cas pourrait être résolu en confrontant plusieurs droites lors du RANSAC par exemple, puisqu'on s'attend à ce qu'il y ait plus de points sur l'interface réelle que sur la frontière ombre-lumière sur la pale.

Pour conclure, il semble raisonnable de pouvoir régler certains détails en faisant une analyse plus poussée des cas qui ne marchent pas, et de réussir à appliquer ces méthodes sur le terrain.



FIGURE 17. Exemple de cas où l'angle est bien calculé mais pas la droite.

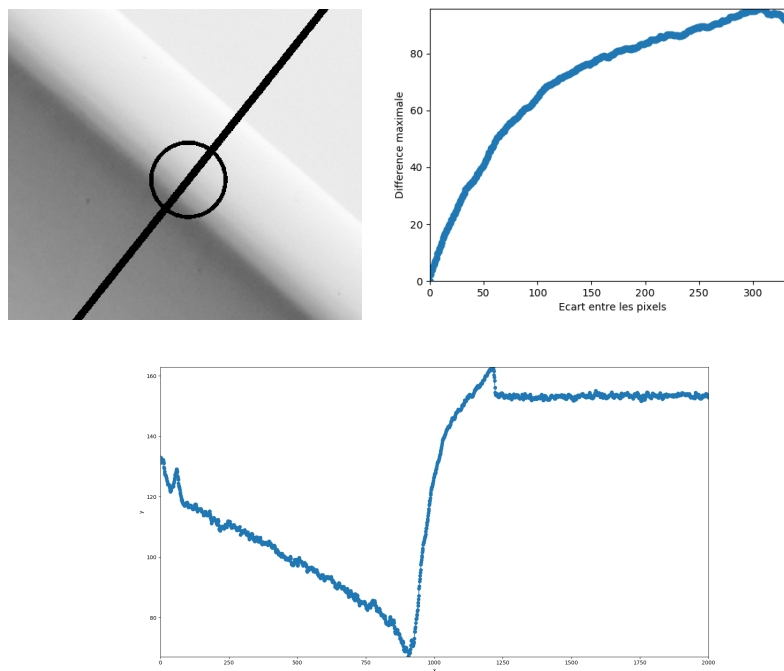


FIGURE 18. Exemple de mauvaise détection du point. Le graphe en bas correspond au niveau de gris sur la ligne considérée. On comprend pourquoi l'algorithme ne fonctionne pas : la longue transition de l'ombre sur la pale masque la transition ciel-pale, plus abrupte mais avec moins de contraste.

A. SAINT DIZIER, MAP5, Université Paris Descartes, 75006, Paris, France

*E-mail* : alexandre.saint-dizier@parisdescartes.fr

A. NICOLOPOULOS, LJLL, Université Pierre et Marie Curie, 75005, Paris, France

*E-mail* : nicolopoulos@ljl11.math.upmc.fr

N. RIANE, LJLL, Université Pierre et Marie Curie, 75005, Paris, France

*E-mail* : nizar.riane@etu.upmc.fr

L. TRUSSARDI, Équipe-projet REO, INRIA, 75012, Paris, France • *E-mail* : lara.trussardi@inria.fr